AI&DS

Final Report
AI&DS TR-1015-02

May, 1983

DISTRIBUTED HYPOTHESIS FORMATION IN DISTRIBUTED SENSOR NETWORKS

FINAL REPORT

Prepared for:

Dr. Barry M. Leiner
DARPA
1400 Wilson Boulevard
Arlington, VA 22209

Prepared by:

C.Y. Chong
S. Mori
E. Tse
R.P. Wishner

Approved for public release; distribution unlimited.

ADVANCED INFORMATION & DECISION SYSTEMS
Mountain View, CA 94040

AI&DS
ADVANCED INFORMATION
& DECISION SYSTEMS

201 San Antonio Circle, Suite 286
Mountain View, CA 94040-1270
(415) 941-3912

Final Report
AI&DS TR-1015-02

May, 1983

DISTRIBUTED HYPOTHESIS FORMATION IN DISTRIBUTED SENSOR NETWORKS

FINAL REPORT

C.Y. Chong
S. Mori
E. Tse
R.P. Wishner

Sponsored by

Defense Advanced Research Projects Agency (DoD)
DARPA Order No. 4272

Under Contract No. MDA903-81-C-0333 issued by Department of Army, Defense Supply Service-Washington, Washington, D.C. 20310

The views and conclusions contained in this document are those of the authors and should not be interpreted as represenging the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A133 849

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Distributed Hypothesis Formation in Distributed Sensor Networks

5. TYPE OF REPORT & PERIOD COVERED
Final Report
July 15, 1981 - May 30, 1983

6. PERFORMING ORG. REPORT NUMBER
AI&DS/TR-1015-02

7. AUTHOR(s)
C.Y. Chong, S. Mori, E. Tse, R.P. Wishner

8. CONTRACT OR GRANT NUMBER(s)
MDA903-81-C-0333

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Advanced Information & Decision Systems
201 San Antonio Cir., Suite 286
Mountain View, CA 94040-1270

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
ARPA Order No. 4272

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

12. REPORT DATE
May, 1983

13. NUMBER OF PAGES
274

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Defense Supply Service - Washington
Room 1D 245, The Pentagon
Washington, D.C., 20310

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Multitarget Tracking
Multitarget Surveillance System
Surveillance Correlation
Distributed Systems
Distributed Multitarget Tracking
Distributed Sensor Networks
Distributed Hypothesis Formation
Distributed Estimation Systems
Information Fusion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents research results on distributed situation assessment in a distributed sensor network (DSN). The area of multitarget tracking and classification has been chosen to investigate issues associated with distributed hypothesis formation and evaluation. A general theory for Bayesian multitarget tracking has been developed. This is used as the basis for specifying the processing architecture at each node in the DSN. Each node contains the Generalized Tracker/Classifier for processing of
(Continued)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) cont

local sensor data, an information fusion module to integrate processed information from various nodes, and an information distribution module. The problem of removing redundant information in a general distributed estimation system has also been investigated. Simulation results to study various issues associated with distributed situation assessment are presented.

TABLE OF CONTENTS

Section Page

LIST OF FIGURES

LIST OF TABLES

Table Page

ACKNOWLEDGMENTS

The authors wish to acknowledge the help of Ms. Teena Randall in typing the Final Report and Dr. Steve Spain for editorial suggestions.

1. INTRODUCTION AND SUMMARY

This report describes research on the distributed processing of sensor data for situation assessment at the processing nodes in a distributed sensor network (DSN). This research has been performed at Advanced Information & Decision Systems under the contract entitled "Distributed Hypothesis Formation in Distributed Sensor Networks".

Distributed sensor networks have many positive attributes such as improved performance, faster response time, more flexible communication, and less vulnerability as compared with centralized or hierarchical systems. As a result they are attractive for many Department of Defense applications. These DSNs can consist of a variety of sensor types (e.g., microwave radar, SIGINT, IR, etc.) and are relevant to a variety of defense systems (e.g., air defense, land warfare, space defense, etc.). However, many research issues need to be addressed before such DSN systems can be designed, built and achieve their military potential. In this project we have addressed and resolved some of these issues. This final report summarizes the results of our investigation.

1.1 PROJECT OBJECTIVES AND TECHNICAL APPROACH

The overall objective of this research project was to advance the state of the art in distributed situation assessment in distributed sensor networks. Specifically, [illegible] set out to accomplish the following goals:

- investigate techniques of hypothesis representation, formation and evaluation, etc. in distributed sensor networks;
- investigate various tradeoffs such as computation versus communication, and the performance of centralized, decentralized and distributed structures as a function of various parameters.

The basic system model consists of a distributed system of nodes which are connected in a packet switching network. Each node contains a processor and one or more sensors, whose coverage may overlap those of sensors at other nodes. The input information at each node consists of:

- own sensor data
- messages from other nodes
- contextual information

Our àpproach has been to understand the main technical issues associated with a DSN by concentrating on the hypothesis representation, formation, and evaluation processes. The tracking and classification of multiple targets in low signal-to-noise ratio and high clutter environment was chosen as the particular application area. Our rationale for concentrating on hypothesis representation, formation, and evaluation was that in a dense target environment, with a low detection probability and high false alarm rates, successful tracking and classification of targets depends very much on forming the correct (data association) hypotheses. Thus, multitarget tracking and classification provides a rich problem domain to study distributed problems. In addition, such problems have many applications in the defense area.

We have adopted an approach which is both analytical and heuristic. A DSN is primarily an engineered information gathering and processing system. Detailed mathematical models of sensors, targets, and the environment are usually available. They are used to provide inputs for generating optimal algorithms (in a precise mathematical sense) for processing the data at the nodes. These algorithms are, however, only applicable under ideal situations when there are no computation and/or communication constraints. In a more realistic environment, where the DSN is supposed to operate, these constraints cannot be ignored. They are incorporated into the algorithms by the use of heuristics. In particular, picking the right hypothesis can be regarded as a tree search problem. Some artificial intelligence (AI) based ideas are used to generate the heuristics for managing the search. Although AI provides useful tools for handling distributed hypothesis formation problems, its role in this project has been limited to hypothesis management. This limitation was due to the fact that a mathematical foundation for distributed multitarget tracking was still lacking and had to be developed before the consideration of more complicated issues.

1.2 TECHNICAL ISSUES

There are many technical issues associated with distributed systems. The following are the ones which are particularly relevant to using a DSN for multitarget tracking and classification:

- Local situation assessment

  - how to represent and form hypotheses

  - how to evaluate the goodness of a hypothesis

  - how to keep the growing number of hypotheses within the computational constraints

- Communication level

  - what to communicate
    - data versus hypotheses
    - information to accompany hypotheses

  - when to communicate

  - how to integrate the incoming information into the local information
    - how to process incoming information
    - how to avoid redundant processing of the same information

These are some of the issues which need to be addressed before a DSN can be designed. We have studied these issues in the project, both analytically and experimentally through computer simulations.

1.3 PROJECT ACCOMPLISHMENTS

Little had previously been done on distributed multitarget tracking and classification. Even in the centralized case, the existing results are not good enough to provide a sound foundation for developing distributed algorithms. To supply this foundation, we have developed a theory for centralized multitarget tracking and classification. The resulting algorithm, called the Generalized Tracker/Classifier (GTC), provides a definitive treatment of hypothesis formation, evaluation, and management in a centralized processing system. This theory addresses most of the technical issues associated with local data processing for situation assessment.

The centralized multitarget tracker and classifier has been used to develop algorithms for processing the incoming information at a node and integrating it with the local information. The resulting algorithm, called the Distributed Generalized Tracker/Classifier, prescribes the appropriate processing architecture at a node and represents a systematic treatment of distributed multitarget tracking. Each processing node consists of three modules: the Generalized Tracker/Classifier for processing the local sensor data, an information fusion module to handle the information from other nodes, and an information distribution module for transmission of messages to other nodes. Many issues dealing with information integration at a node have been addressed. One of the most important has been the development of ways to avoid redundant use of the same information at a node.

The algorithms have been coded and simulations have been performed for various distributed scenarios to resolve the issues dealing with the trade-off of communication versus computation. We have discovered that there is a delicate trade-off. Because the number of hypotheses tends to grow rapidly as the amount of data increases, having more data, as in a centralized situation, is not necessarily better unless resources are available to process the data. In general, the quality of the information is more important than the amount of data. With a proper distributed algorithm, performance similar to that of the centralized scheme can be achieved.

1.4 REPORT ORGANIZATION

Results obtained earlier in the project have been documented in an interim technical report and several papers. The interim technical report [1] and the paper [2] contain a description of the centralized Generalized Tracker/Classifier. A summary of the overall project has been reported in [3], and [4] describes a framework for the general distributed estimation problem.

The rest of the report is organized as follows. In Section 2 we describe the system and the major components at each processing node in the network. The three modules are the Generalized Tracker/Classifier (GTC), the information fusion module and the information distribution module. Section 3 describes the Generalized Tracker/Classifier (GTC), which carries out the processing of local sensor data at each processing node. The GTC also defines hypothesis representation, hypothesis formation, and hypothesis evaluation for general multitarget tracking and classification problems. The information fusion module in a distributed GTC is described in Section 4; it contains submodules which are analogous to those of the GTC except they deal with processed information from various nodes instead of sensor data. Section 5 considers information distribution and problems associated with general estimation problems in a network. An example illustrates some pitfalls which can result from careless information processing for a network. Two numerical examples are described in Section 6 to illustrate the tradeoffs of computation versus communication and their effects on system performance. Specifically, the performance of centralized, decentralized and distributed systems as a function of various parameters is considered. Section 7 contains conclusions and suggestions for future research.

Three appendices contain the details of the algorithms described in the main body of the report. Appendix A is a paper on the theory behind the centralized Generalized Tracker/Classifier. Appendix B presents the theory on distributed estimation over a network. In particular, it gives the distributed fusion formula for each node in a network. Appendix C describes a theory for distributed multitarget tracking and classification. This theory serves as the basis for the information fusion algorithms used in each processing node.

2. SYSTEM DESCRIPTION AND NODAL ARCHITECTURE

2.1 SYSTEM DESCRIPTION

In this section we describe the structure of the system under consideration. The distributed sensor network (DSN) consists of a collection of processing nodes, each with one or more sensors and a communication network which connects the processing nodes. The structure of the system is shown in Figure 2-1.

Each sensor generates measurements from the targets which are within its field-of-view. The sensors are supposed to be generic and not of a particular type. They have the following characteristics:

1. The probability of detection of the targets by a sensor is less than one and depends on the relative positions of the targets to the sensor. If the target is not within the sensor's field-of-view, it will not be detected. For certain types of sensors, such as the MTI radar, only targets whose radial velocities with respect to the sensors lie above a certain threshold are detected.
2. False alarms are generated and correspond to ground clutter, etc. The reports from the sensors may contain (discrete) feature measurements as well as the usual (continuous) measurements such as position and velocity.

Each processing node collects measurements from a set of sensors. It is convenient to assume that the sensor sets for different processing nodes are disjoint. The function of each processing node is to process

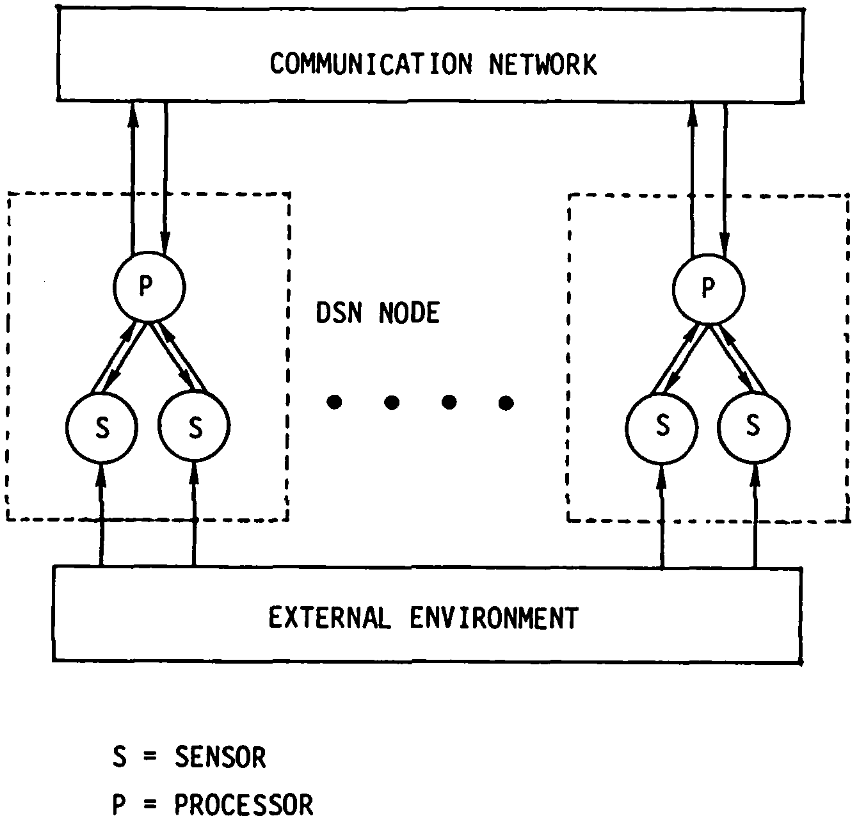

Figure 2-1: Distributed Sensor Network

the local sensor data to form an assessment of the state of the world, to distribute information to other nodes, and to combine the information obtained from other nodes with the local information to update its assessment about the state of the world. The processing nodes are thus the main information processing units in the DSN.

The communication network communicates messages from one processing node to the other processing nodes. The actual network may be a packet radio or other kind of networks. Our study has not gone into any details on the communication network but has only considered it as a means of allowing certain nodes to share information.

2.2 STRUCTURE OF EACH PROCESSING NODE

The purpose of each processing node is to integrate the data from local sensors with information from other nodes to form an assessment of the state of the world. There are three generic functions of the processing node.

- processing of the local sensor data
- information fusion from other nodes
- information distribution to other nodes

These three functions are implemented as three separate modules within each processing node. The structure of each node in the system is shown in Figure 2-2. The three modules are discussed briefly below and in detail in Sections 3, 4 and 5.

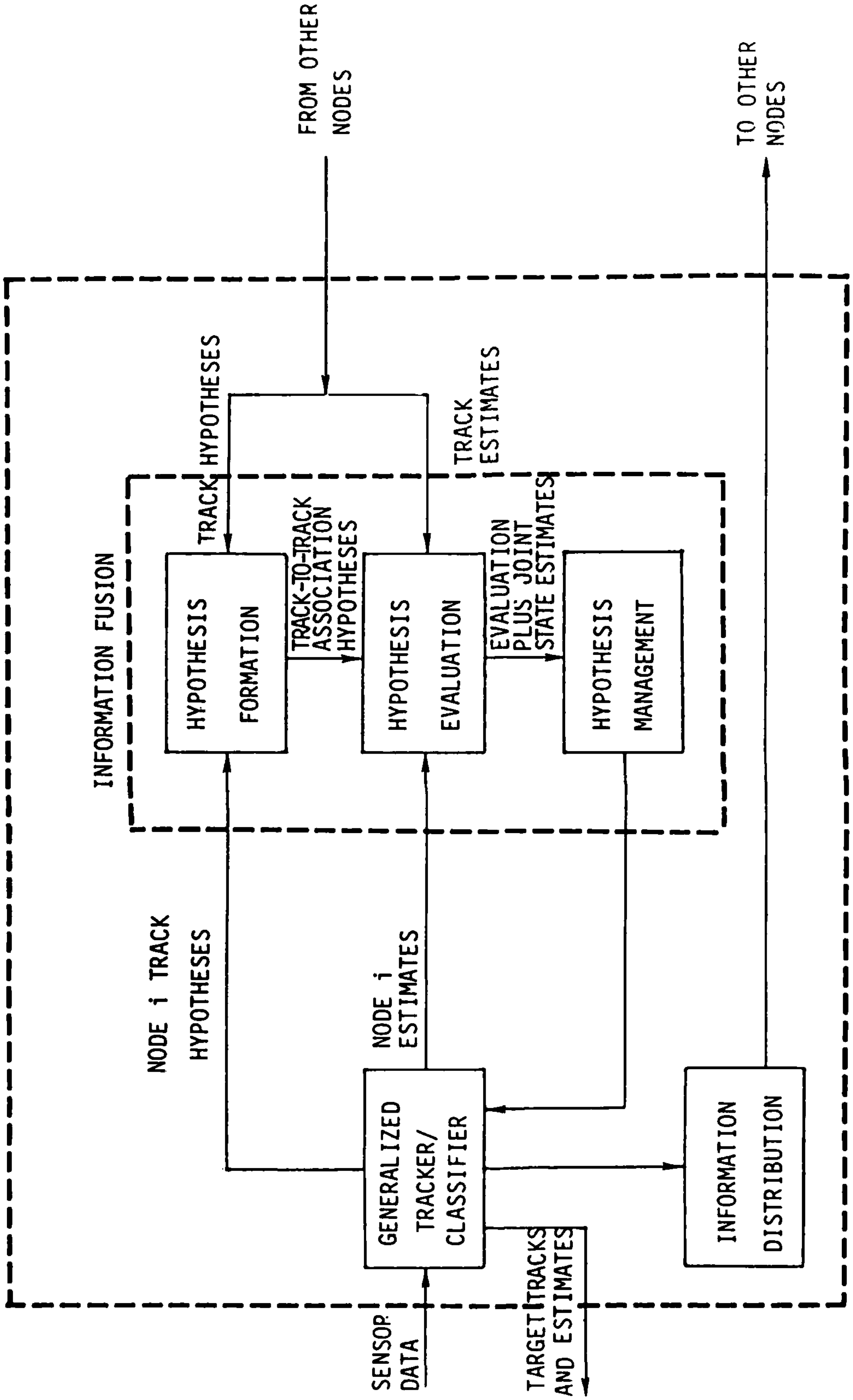

Figure 2-2: Structure of Processing Node

2.2.1 Local Processing of the Sensor Data

This function is responsible for the local data processing before any communication with the other nodes is carried out. Since the objective of the system under consideration is the tracking and classification of multiple targets, this function will be a multitarget tracker. In our system, it is called the Generalized Tracker/Classifier (GTC). The GTC forms multiple hypotheses, each consisting of a collection of tracks to explain the origins of the measurements in each data set. These hypotheses are then evaluated with respect to their probabilities of being true. To stay within the computational constraints of each node, the hypotheses are pruned, combined, clustered, etc. The result of this processing is a set of hypotheses and their probabilities, a collection of tracks corresponding to possible targets and the state distributions of these tracks. These quantities together constitute the information state for multitarget tracking.

2.2.2 Information Fusion

This module combines the local information with information obtained from the other nodes to obtain a new assessment. The information from the local nodes consists of the information described above. The information from other nodes is also similar. Information fusion then consists of the following steps:

1. Hypothesis Formation - Given a set of hypotheses from other nodes, this submodule generates new global hypotheses. Tracks from the hypotheses of different nodes are associated in all possible ways,

whether they correspond to the same or different targets.

2. Hypothesis Evaluation - Each of the hypotheses formed above is then evaluated with respect to its probability of being true. The statistics of the tracks from different hypotheses are used in this evaluation. For example, if two tracks are widely apart in their position or velocity distributions, they are more likely to have come from different targets than the same target.

3. Hypothesis Management - This is again needed to make computation feasible within the available resources.

2.2.3 Information Distribution

This module decides what information is to be transmitted, who gets the information, and when it should be communicated. It thus specifies the information available to each node at any time, i.e., the information structure of the system.

3. GENERALIZED TRACKER/CLASSIFIER (GTC)

In this section, we describe the Generalized Tracker/Classifier (GTC) which is the module for processing the local sensor data within each node in the DSN. The GTC structure is shown in Figure 3-1 and the theory upon which it is based has been described in more detail in an earlier report [1] and in Appendix A. A summary can also be found in [2]. The GTC represents the most complete theory thus far available for Bayesian multitarget tracking and classification. It can be shown, as in Appendix A, that many existing algorithms are special suboptimal cases of the GTC when the appropriate approximations are made. In addition, the GTC can handle complex situations such as targets moving as a group and state dependent detection probabilities which are not considered in the existing algorithms. We shall first describe the models used in the GTC and then the modules in the actual tracker.

3.1 TARGET AND SENSOR MODELS

This section describes the target and sensor models. These models provide the mathematical foundation for the modules of the tracker.

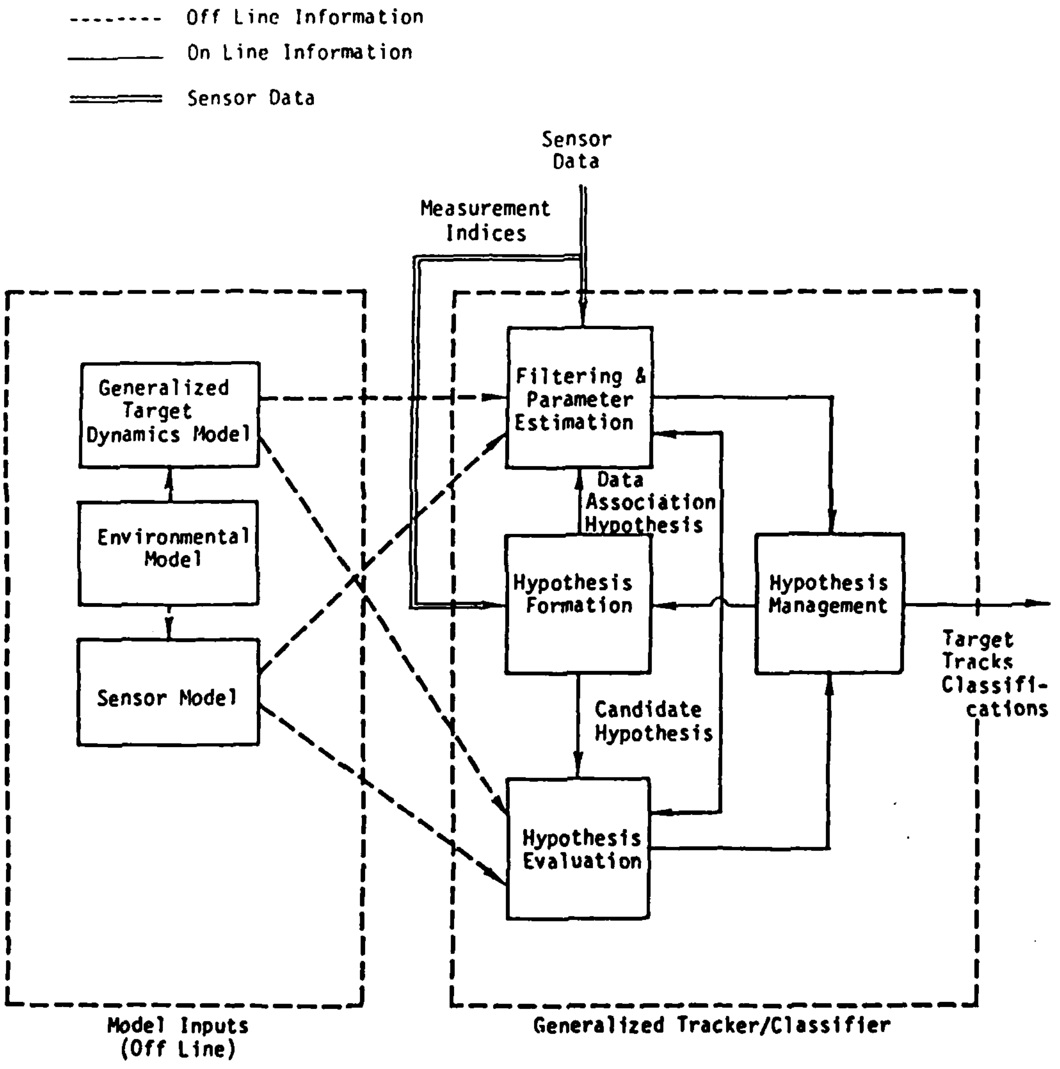

Figure 3-1: Generalized Tracker/Classifier

3.1.1 Target Model

A novel feature of the target model used in our approach is that the targets are modeled together as one target system state. More specifically, the <u>target</u> <u>system</u> <u>state</u> at any time t is $(X(t), N_T(t))$ where N_T is the number of targets and $X(t)$ is the composite state of all targets. Information about the total number of targets is very useful in multitarget tracking. For example, if it is known that the maximum number of targets is 10, any sensor report containing more than 10 measurements would most likely (unless there are split measurements) contain some false alarms. In the current approach, knowledge on the number of targets is viewed as an integral part in the target model. N_T can have arbitrary probabilistic descriptions, but a particularly useful assumption is that N_T is a constant and has a Poisson distribution with mean ν_o. Thus,

$$\text{Prob.}\{N_T = n\} = \frac{\nu_o^n}{n!} \exp(-\nu_o). \tag{3.1}$$

Given $N_T(t) = n$, the composite state for the n targets in general consists of two parts: a part corresponding to the common target state such as the group position, velocity or type if we are dealing with a group, and a part corresponding to the individual target states. This structure allows us to handle complex target structures such as targets moving as a group.

In many applications, the group part is absent and the individual target models are independent and identically distributed random processes. The state of the i^{th} target, x_i, is then characterized by the initial distribution/density

$$\text{Prob.}\{x_i(t_o) \in dx\} = q_o(x)\mu(dx) \tag{3.2}$$

and transition probability density

$$\text{Prob.}\{x_i(t+\Delta t) \in dx|x_i(t) = x'\} = f_{\Delta t}(x|x')\mu(dx). \tag{3.3}$$

In general, $x_i(t)$ is an element in a hybrid set X, where the continuous part corresponds to position, velocity, etc., and the discrete part corresponds to the type of targets, sudden structural changes in dynamics (maneuvering targets), changes in operational modes, etc. μ is the hybrid measure on X, i.e., the direct product of the usual Lebesgue measure for the continuous state space and the counting measure for the discrete state set. The usual linear continuous models assumed in multitarget tracking are then special cases of this model. For the rest of this report, we consider the case where no group information is available and the target models are independent and identically distributed random processes.

3.1.2 Sensor Model

At a scan or observation time t, a sensor s generates a data set $((y_j)_{j=1}^{N_M}, N_M, t,s)$ where N_M is the number of measurements in the data set, and y_j is the j^{th} measurement in the set. Each y_j takes a value in the measurement value space $\mathcal{Y}_s$ for the sensor s. $\mathcal{Y}_s$ is, in general, a hybrid set with measure μ_s. The continuous part of $\mathcal{Y}_s$ corresponds to an observed position or velocity, etc., while the discrete part corresponds to observed features.

Given a target system state $((x_i(t))_{i=1}^{N_T}, N_T)$, the generation of a data set is characterized by the following four steps:

- Target Detection: The set of detected targets by sensor s at time t is a random subset $I_D(t,s)$ of the target index set $\{1,\ldots,N_T\}$. It can be characterized by the detection function $F_D(t,s)$ which is the random indicator function of $I_D(t,s)$, i.e., $F_D(t,s)(i) = 1$ if $i \in I_D(t,s)$ (target i is detected) and $F_D(t,s)(i) = 0$ otherwise (target i is not detected).

  We assume that

$$\text{Prob.}\{F_D(t,s) = 1|x_i(t)\} = p_D(x_i(t)|t,s) \tag{3.4}$$

  with a common detection probability function $p_D(.|t,s)$. Thus the detection of a target i is a conditionally independent event determined only by the target state $x_i(t)$, the sensor s and the time t. The number of targets detected by sensor s at time t is given by

$$N_D(t,s) = \#(I_D(t,s)) \tag{3.5}$$

where # denotes the number of elements in a set.

- Number of False Alarms: The number of false alarms, $N_{FA}(t,s)$, generated by sensor s at time t depends only on the time and the sensor and is independent of the target state or any other sensor data. Specifically, its probability is given by

$$\text{Prob.}\{N_{FA}(t,s)\} = p_{N_{FA}}(N_{FA}(t,s)|t,s). \tag{3.6}$$

The total number of measurements in the data set is then given by

$$N_M(t,s) = N_D(t,s) + N_{FA}(t,s). \tag{3.7}$$

Let $J_M(t,s) = \{1,\ldots,N_M(t,s)\}$ be the set of measurement indices from sensor s at time t

- Measurement Random Assignment: Given the set $I_D(t,s)$ of detected targets, the measurement indices of the detected targets are modeled by the assignment function A(t,s). This is a random function defined on $I_D(t,s)$ and taking values in $J_M(t,s)$ such that for each $i \in I_D(t,s)$ and $j \in J_M(t,s)$,

$$j = A(t,s)(i) \tag{3.8}$$

means that the j^{th} measurement originates from target i. A realization of A(t,s) is an one-to-one mapping from $I_D(t,s)$ to $J_M(t,s)$. We assume any particular order of measurements in the data set does not contain any information about the targets, i.e., it is completely random. Thus, given $I_D(t,s)$ and $J_M(t,s)$, any realization of A(t,s) is equally likely and its probability is given by

$$\text{Prob.}\{A(t,s)|N_M(t,s),\ I_D(t,s),\ X(t),\ N_T\} = \frac{(N_M(t,s)-N_D(t,s))!}{N_M(t,s)!}$$

$$= \frac{N_{FA}(t,s)!}{N_M(t,s)} \qquad (3.9)$$

- Measurement Values: Given the set of detected target indices $I_D(t,s)$, the set of measurement indices $J_M(t,s)$ and the random assignment function A(t,s), the measurement (value) $y_{A(t,s)(i)}$ originating from a detected target i is conditionally independent of the other measurement values and depends only on the target state $x_i(t)$, the sensor s and the time t, i.e.,

$$\text{Prob.}\{y_{A(t,s)(i)} \in dy|A(t,s),\ X(t),N_T\}$$

$$= p_m(y_{A(t,s)(i)}|x_i(t),t,s)\mu_s(dy) \qquad (3.10)$$

where $p_m(.|x_i(t),t,s)$ is the common probability density function.

Given the set $J_{FA}(t,s)$ of false alarms, each measurement value y_j for the j^{th} false alarm in $J_{FA}(t,s)$ is completely independent (of each other and of the targets) and has a common probability density $p_{FA}(y_j|t,s)$.

The main feature of this sensor model is its generality. The number of measurements, as well as the actual measurement values themselves, are considered an integral part of the sensor report. Furthermore, $p_D(x|t,s)$, the detection probability of a target, not only depends on the time t and the sensor s, but is also a function of the state of the target. This state dependence is particularly useful when there is masking of the targets or when sensor detection depends on the radial velocity of the target, as in a MTI radar.

3.2 HYPOTHESIS FORMATION

Hypothesis formation is the first step in the GTC operation. It forms the feasible associations of data from different times and different sensors.

3.2.1 Tracks and Hypotheses

Since the origins of the measurements in each sensor report or data set are uncertain, one of the crucial steps in multitarget tracking is the formation of the data-to-data association hypothesis, or simply the hypothesis. Each hypothesis corresponds to a possible explanation of the origins of the measurements. These hypotheses would then be evaluated with respect to their probabilities of being true in the later steps.

We can index each data set by k = (t,s), the time t when it is generated and the sensor s reporting the data. Each data set $((y_j)_{j=1}^{N_M}, N_M, t,s)$ can then be represented by z(k). Let $\underline{K}$ be the collection of all the data set indices, called the data set index set. The order in which the data sets arrive define a natural lexicographic order $\leq$ on $\underline{K}$.

Consider the cumulative measurement index set at time k defined as

$$J_M^{(k)} = \bigcup_{k' \leq k} J_M(k') \times \{k'\}.$$

Each element (j,k) = (j,t,s) in $J_M^{(k)}$ represents the j^{th} measurement in the data set from sensor s at time t. Our objective is to explain the origin of each of these elements.

According to the sensor model, the uncertainty in the measurements for the detected targets is due to the random assignment A(k). For each target i, the set of measurement indices originating from i is given by

$$T(i) = \{(A(k)(i),k) | k \in \underline{K}, i \in I_D(k)\}. \qquad (3.11)$$

Since $I_D(k)$ and A(k) are random, each T(i) is a random set. Let

$$\begin{aligned} \Lambda &= \{T(i) | i \in \bigcup_{k \in K} I_D(k)\} \\ &= \{T(i) | i \in \{1,\ldots,N_T\}, T(i) \neq \phi\} \qquad (3.12) \end{aligned}$$

Then Λ is a random set identifying the measurement for all the detected targets. At any k, the restriction of Λ to k is defined as

$$\Lambda_{|k} = \{T \cap J_M^{(k)} | T \in \Lambda, T \cap J_M^{(k)} \neq \phi\}, \qquad (3.13)$$

which identifies the measurement indices for each of the targets detected up to k.

For each target i, T(i), its measurement indices, has realized values which are subsets of $J_M^{(k)}$. Each of these subsets is called a <u>track</u> at k and denoted by τ. Since we assume that a target can generate at most one measurement in a data set, then the set of possible tracks are those containing at most one measurement index from each data set. Let the set of possible tracks at k be $\mathcal{T}(k)$.

A <u>data-to-data association hypothesis</u> λ (henceforth called a hypothesis) at k is a (possibly empty) collection of non-empty tracks. A hypothesis λ is thus a particular realization of the random set $\Lambda_{|k}$. Again, since we assume no merged measurements, then in the set of possible hypotheses, each hypothesis cannot have intersecting tracks. Let $\mathcal{H}(k)$ be the set of all possible hypotheses at k.

Thus, $\mathcal{H}(k)$ is the set of all possible realizations for the random set $\Lambda_{|k}$, i.e., $\mathcal{H}(k)$ consists of all possible explanations of the origins of the measurements in the data sets up to k. An event $\{\Lambda_{|k} = \lambda\}$ means that:

1. $\#(\lambda)$ targets have been detected and included in at least one of the data sets prior to and including k;

2. each track τ in λ corresponds uniquely to a target detected prior to or at k;

3. for any $k' \leq k, (j',k') \in \tau$ implies that the j^{th} measurement in the data set k' originates from the target identified by τ;

4. if the intersection of τ and $J_M(k') \times \{k'\}$ is empty, then the target is falsely dismissed in k';

5. any measurement indices not in $\bigcup \lambda$ are false alarms.

Similar to (3.13) for each k in $\underline{K}$ and each $k' \leq k$, we can define the restriction of τ to $J_M^{(k')}$ (or simply to k') as

$$\tau' = \tau_{|k'} = \tau \cap J_M^{(k')}. \qquad (3.14)$$

The restriction of λ to k' can be defined similarly as

$$\lambda' = \lambda_{|k'} = \{\tau_{|k'} | \tau \in \lambda\} \backslash \{\phi\}. \qquad (3.15)$$

In the above τ' and λ' are called the predecessors of τ and λ, respectively.

3.2.2 Tree Representation

Although the hypotheses are defined as collections of tracks which are in turn collections of measurement indices, an equivalent representation is by means of a tree. In this representation, each level in the tree corresponds to a measurement index and a node corresponds to a target. Hypothesis formation given a new sensor report then reduces to the expansion of the tree and a branch of the tree represents a particular data-to-data association hypothesis. The concept of the predecessor of a hypothesis is obvious from this representation.

Figure 3-2 shows a hypothesis tree for two data sets with two measurements in each. The tracks associated with each hypothesis are also given. For example, hypothesis 24 associates y_1^1 and y_1^2 with the same target (track 5), and y_2^1 with a different target (track 2). It thus hypothesizes y_2^2 to be a false alarm. Note that from two data sets with two measurements each, we have eight possible tacks and a total of 34 possible hypotheses.

3.3 HYPOTHESIS EVALUATION

Many data-to-data association hypotheses are generated by the Hypothesis Formation Module. In order to rank these hypotheses, the Hypothesis Evaluation Module evaluates the probability of each hypothesis. This evaluation is based on the target models, sensor models and the measurement values. For general target models, the evaluation formula can be found in Appendix A. The following describes

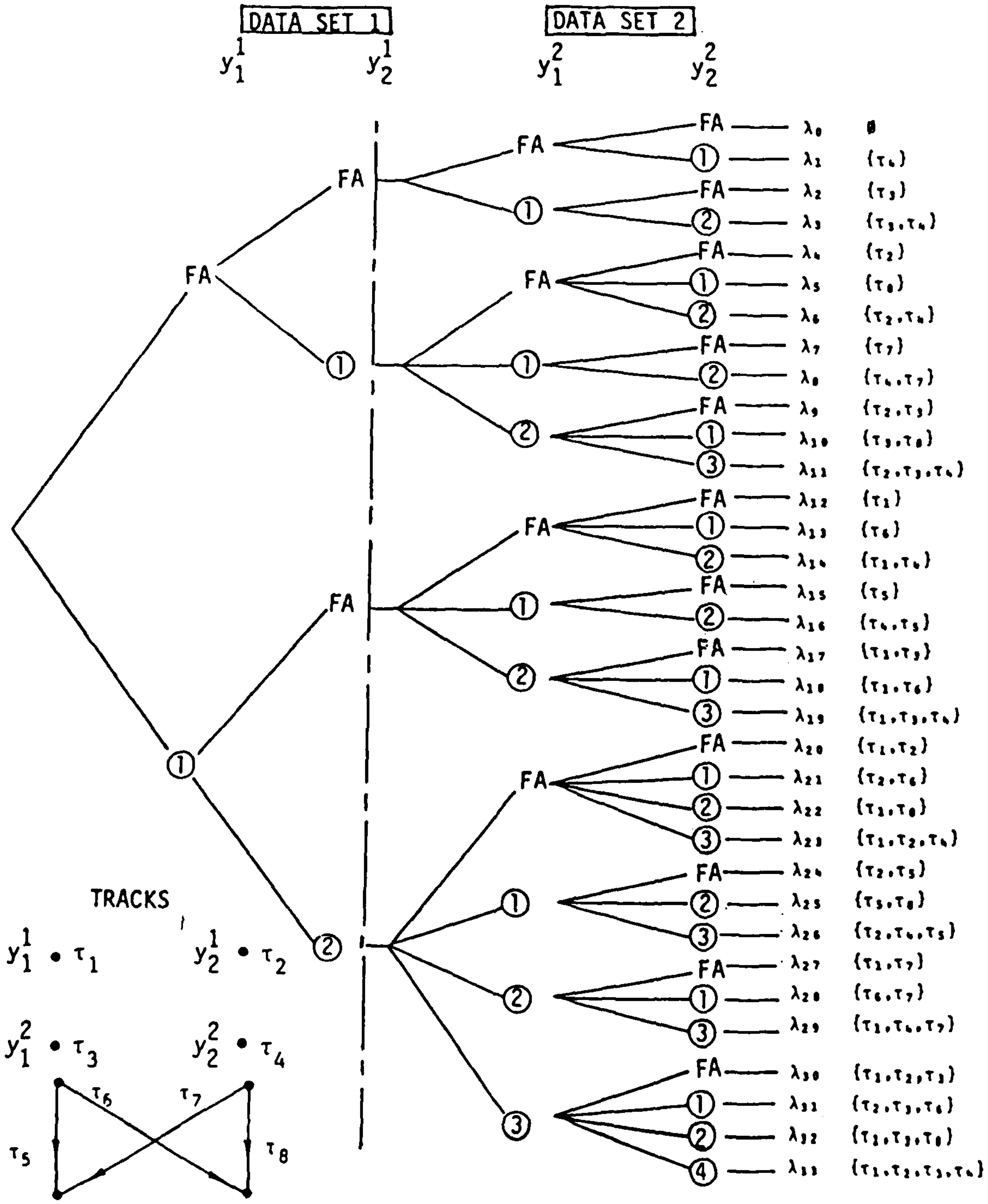
DATA SET 1
DATA SET 2
y_1^1
y_2^1
y_1^2
y_2^2
FA
①
②
③
④
λ_0 ∅
λ_1 $\{\tau_4\}$
λ_2 $\{\tau_3\}$
λ_3 $\{\tau_3,\tau_4\}$
λ_4 $\{\tau_2\}$
λ_5 $\{\tau_8\}$
λ_6 $\{\tau_2,\tau_4\}$
λ_7 $\{\tau_7\}$
λ_8 $\{\tau_4,\tau_7\}$
λ_9 $\{\tau_2,\tau_3\}$
λ_{10} $\{\tau_3,\tau_8\}$
λ_{11} $\{\tau_2,\tau_3,\tau_4\}$
λ_{12} $\{\tau_1\}$
λ_{13} $\{\tau_6\}$
λ_{14} $\{\tau_1,\tau_4\}$
λ_{15} $\{\tau_5\}$
λ_{16} $\{\tau_4,\tau_5\}$
λ_{17} $\{\tau_1,\tau_3\}$
λ_{18} $\{\tau_1,\tau_6\}$
λ_{19} $\{\tau_1,\tau_3,\tau_4\}$
λ_{20} $\{\tau_1,\tau_2\}$
λ_{21} $\{\tau_2,\tau_6\}$
λ_{22} $\{\tau_1,\tau_8\}$
λ_{23} $\{\tau_1,\tau_2,\tau_4\}$
λ_{24} $\{\tau_2,\tau_5\}$
λ_{25} $\{\tau_5,\tau_8\}$
λ_{26} $\{\tau_2,\tau_4,\tau_5\}$
λ_{27} $\{\tau_1,\tau_7\}$
λ_{28} $\{\tau_6,\tau_7\}$
λ_{29} $\{\tau_1,\tau_4,\tau_7\}$
λ_{30} $\{\tau_1,\tau_2,\tau_3\}$
λ_{31} $\{\tau_2,\tau_3,\tau_6\}$
λ_{32} $\{\tau_1,\tau_3,\tau_8\}$
λ_{33} $\{\tau_1,\tau_2,\tau_3,\tau_4\}$
TRACKS
y_1^1 • τ_1
y_2^1 • τ_2
y_1^2 • τ_3
y_2^2 • τ_4
τ_5
τ_6
τ_7
τ_8

Figure 3-2: Hypothesis Tree

the evaluation scheme for independent and identically distributed target models, which are the emphasis in this report.

Let $Z^{(k)}$ be the cumulative data set at k, i.e., $Z^{(k)} = \{z(k')|k' \leq k\}$, and λ be a hypothesis defined on $J_M^{(k)}$. We would like to evaluate Prob.$\{\Lambda_{|k} = \lambda|Z^{(k)}\} = P(\lambda|Z^{(k)})$. Let k′ be the immediate predecessor of k, the latest data index and assume Prob.$\{\lambda_{|k'}|Z^{(k')}\}$ is known. Suppose $z(k) = (y,m,k)$. Then each hypothesis can be evaluated as

$$P(\lambda|Z^{(k)}) = C^{-1}P(\lambda_{|k'}|Z^{(k')})\, L_{FA}(k,\lambda) \prod_{\tau \in \lambda} (\nu(k'))^{\epsilon(\tau)}\, L_k(Y(\tau,k),\tau) \qquad (3.16)$$

where C is a normalization constant and the L′s are likelihood functions. The False Alarm Likelihood Function is:

$$L_{FA}(k,\lambda) = n_{FA}(\lambda|k)!\; p_{N_{FA}}(n_{FA}(\lambda|k)) \prod_{j \in j_{FA}(\lambda,m|k)} p_{FA}(y_j|k) \qquad (3.17)$$

where $n_{FA}(\lambda|k)$ is the number of false alarms in Z(k) according to λ, and $j_{FA}(\lambda,m|k)$ is the set of false alarms in z(k) according to λ.

The Track-Measurement Likelihood Function is:

$$(\nu(k'))^{\epsilon(\tau)}\, L_k(Y(\tau,k),\tau) \qquad (3.18)$$

where

$\nu(k')$ = the expected number of targets which are not detected up to or at k',

$\epsilon(\tau)$ = 1 if τ is a new track in z(k); 0 otherwise,

$Y(\tau,k)$ = the measurement for track τ in the data set k; (if track τ is missed in k, $Y(\tau,k) = \theta$)

$$L_k(y,\tau) = \int_X g_k(y|x)\bar{p}_\tau^{(k)}(x)\mu(dx) \tag{3.19}$$

$\bar{p}_\tau^{(k)}(x) = p(x(t)|Z(k'),\tau)$ is the density of x(t) given Z(k') and track τ, and

$$g_k(y|x) = \begin{cases} p_M(y|x,k)p_D(x|k) & \text{if } y \neq \theta \\ 1-p_D(x\ k) & \text{if } y = \theta \end{cases} \tag{3.20}$$

There are thus three types of track-measurement likelihood functions to be evaluated.

1. the likelihood $L_k(Y(\tau,k),\tau)$ of measurement $Y(\tau,k) \neq \phi$ originating from a previously detected target $(\tau_{|k'} \neq \phi)$;

2. the likelihood $L_k(Y(\tau,k),\tau)$ of a previously detected target $(\tau_{|k'} \neq \phi)$ being undetected $(Y(\tau,k) = \theta)$ and,

3. the likelihood $\nu(k')L_k(Y(\tau,k),\tau)$ of a measurement $Y(\tau,k) \neq \theta$ originating from a newly detected target $(\tau_{|k'} = \phi)$.

In addition, $L_k(\theta,\phi)$, the likelihood of an undetected target remaining undetected is also used. The evaluation formula is very general and reduces to the standard ones used in multitarget tracking when the appropriate approximations are made.

3.4 FILTERING AND PARAMETER ESTIMATION

This module generates all the necessary statistics used in hypothesis evaluation. In particular, if $\bar{p}^{(k)}(x)$ is the (prior) probability density/distribution of x(t) given $Z^{(k')}$ and τ, and $p^{(k)}(x)$ is the (posterior) probability density/distribution of x(t) given $Z^{(k)}$ and τ, then for every $\tau \in \mathcal{T}^{(k)}$,

$$\bar{p}^{(k)}(x) = \begin{cases} \int_{\mathcal{X}} f_{t-t'}(x|x')p_\tau^{(k')}(x')\mu(dx') & \text{if k has an immediate predecessor } k' = (t',s') \\ \int_{\mathcal{X}} f_{t-t_o}(x|x')q_0(x')\mu(dx') & \text{otherwise.} \end{cases} \quad (3.21)$$

$$p_\tau^{(k)}(x) = L_k(Y(\tau,k),\tau)^{-1}\, g_k(Y(\tau,k)|x)\bar{p}_\tau^{(k)}(x) \quad (3.22)$$

In addition, the expected number of undetected targets can be updated via

$$\nu(k) = \begin{cases} \nu(k')L_k(\theta,\phi) & \text{if k has an immediate predecessor } k' \\ \nu_0 L_k(\theta,\phi) & \text{if k is the minimum in } \underline{K}. \end{cases} \quad (3.23)$$

3.5 HYPOTHESIS MANAGEMENT

The hypothesis management module controls the growth in the number of hypotheses and makes the implementation of the GTC feasible. It is model-independent in the sense that the techniques involved are applicable to a wide class of scenarios. The user, however, should select certain parameters to conform with the computational and memory requirements or to reflect his knowledge about the complexity of the situation.

No general theory on hypothesis management techniques exists at the present moment. The purpose of this section is to summarize some existing techniques and describe any modifications to such techniques that have been adopted in our work. We divide the hypothesis management techniques into the following four classes.

1. Pruning .... cutting branches
2. Combining .... binding branches together
3. Windowing .... data validation
4. Clustering .... data partition

In the following, we discuss the techniques according to the above classification. Although some of the techniques described below may apply to the general target models, for the most part we restrict our attention to the i.i.d. target case.

3.5.1 Pruning

Pruning techniques can be further classified into (1) thresholding, (2) breadth control, and (3) adaptive pruning techniques.

The basic philosophy behind thresholding is to cut (or remove) the "insignificant" hypotheses which in turn tend to produce more insignificant hypotheses. In [5], it is proposed to cut any hypothesis with posterior probability less than a fixed predetermined threshold. In the i.i.d. target case, thresholding may be performed at the track level using the track likelihood functions. One of the disadvantages of this fixed thresholding is that it is performed without considerating the available computational resources or the external condition (e.g., clear versus confusing, etc.). For example, given the same computational resources, one should be able to keep more hypotheses for a small amount of data than for a large amount of data. This adaptivity, however, is not present in the fixed thresholding scheme.

This consideration leads to the second subclass of breadth control techniques in which a fixed number, say M, of the best hypotheses are chosen and propagated forward. This technique is proposed by Keverian in [6]. Choosing a fixed breadth M makes sense when we regard the number of hypotheses kept as a measure of the computational and memory requirements. However, fixed breadth control may deviate from its original rationale quickly when some form of clustering is used since the resources cannot be efficiently allocated among the clusters. Also, the breadth control or fixed breadth method requires a sorting algorithm which requires additional effort (although this issue may not be important since many good sorting algorithms do exist). When breadth control is used extensively to its limits, only one (best) hypothesis is selected and propagated forward. In [5], this model of pruning is called a zero-scan algorithm.

Although fixed-threshold pruning may be viewed as adaptive breadth-control pruning, and vice versa, these techniques do not really adapt to the complexity of the situation. The third subclass of pruning techniques introduced in this project is called adaptive pruning. In this strategy, the hypotheses are first sorted in descending order of their posterior probabilities. Then, when the cumulative sum of the probabilities exceeds a given threshold, the remaining low probability hypotheses are pruned. This method may be called adaptive-threshold/adaptive-breadth pruning since it adjusts both the absolute threshold and the breadth according to the complexity of the external condition, i.e., the more complex the situation is the more low probability hypotheses are retained. This adaptive pruning technique makes more sense than other pruning methods when clustering is used and may be viewed as a way of automatically allocating computational and memory resources among the clusters. However, it still suffers from the same drawback of any fixed thresholding scheme in that the actual (absolute) computational and memory resources cannot be predicted. Furthermore, some form of sorting is still needed.

From a theoretical point of view, the posterior probabilities of hypotheses may be considered as a discrete distribution. Hypothesis pruning may then be viewed as picking the approximation techniques foı the distribution. Figure 3-3 displays the approximation involved in the three schemes. Thus the theoretical issues in hypothesis pruning are what a good approximation should be and how it influences the future evaluation of multitarget tracking. Although some theories on the approximation of probability distributions (e.g., Sorenson and Alspach [7]) may give us some insight, we believe that hypothesis pruning is still an open research area.

3.5.2 Combining

The existing combining techniques can be divided into two subclasses: (1) distribution-oriented techniques, and (2) measurement-index-oriented techniques.

The philosophy behind the first subclass is to combine two similar hypotheses, where similarity is interpreted in a certain way. According to Reid [5], two hypotheses are similar if they have the same number of tracks and each track in one hypothesis has a unique companion track which is similar to it in the other hypothesis. The similarity of tracks is measured by the state estimate distributions, which accounts for the name of distribution-oriented techniques. The rationale behind this approach is that each track state distribution should reflect all

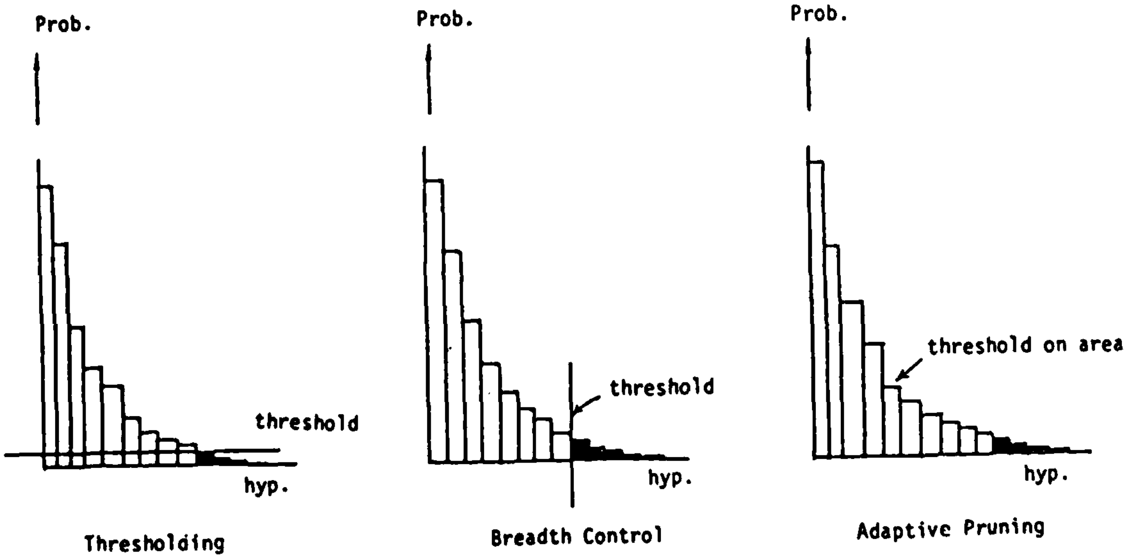

Figure 3-3: Pruning Strategies

the relevant information which affects any future event due to the underlying Markovian assumptions. Thus, if two state distributions of tracks are close enough, we would expect the future behavior of the two tracks to be similar.

Suppose two hypotheses λ_1 and λ_2 are similar, where $\tau_i = \{\tau^i_1, \ldots, \tau^i_n\}$, i = 1, 2. Then hypothesis combining leads to a new hypothesis $\lambda = \{\tau_1, \ldots, \tau_n\}$ with

$$p(\lambda|Z) = p(\lambda_1|Z) + p(\lambda_2|Z) \tag{3.24}$$

and for $j = 1, \ldots, n$

$$p(x|\tau_j, Z) = \frac{p(\lambda_1|Z)p(x|Z,\tau^1_j) + p(\lambda_2|Z)p(x|Z,\tau^2_{\Pi(j)})}{p(\lambda_1|Z) + p(\lambda_2|Z)} \tag{3.25}$$

where $\Pi(.)$ is a permutation which maps a track into a similar track.

However, there still remains the crucial question of choosing a good measure of "similarity" and a good threshold for that measure. When each track distribution is Gaussian, Reid [5] proposes inequality tests using the means and the diagonal elements of the covariance matrices. However, no theoretical justification for the use of those particular inequalities is given. His intuitive reason is that for tracks to be similar, both their means and their variances should not be widely different. This test may work well for Gaussian distributions, but cannot be applied to more general distributions.

Distribution-oriented combining is used to its extreme in [8] and [9] where all the hypotheses are combined after proper windowing (described below). This is only possible when a fixed number of targets are assumed, i.e., every hypothesis has the same number of tracks. When two Gaussian distributions are combined, the combined distribution becomes a Gaussian sum distribution because two different hypotheses represent two disjoint events. When a Gaussian sum distribution is approximated by a Gaussian distribution, the means and the variances are usually equated. Unlike the results in [5] or [8], the Gaussian sum form is preserved to a certain extent in [10] where each track distribution remains a Gaussian sum rather than Gaussian. In this case, the hypothesis trees are extended to include one lower level, namely the distribution level. The hypothesis management (pruning and combining) techniques must then be extended to include this level.

In summary, unless each track distribution is assumed to be and forced to be Gaussian, the similarity criteria proposed in [5], [8], etc., may sometimes be unjustified. Theoretical results on similarity criteria are still lacking in our opinion.

On the other hand, the measurement-index-oriented combining techniques consider each track as a subset of the past cumulative measurement index set. This technique has been proposed by Singer, et. al., [11] and is a classical technique in the multitarget tracking literature. In these schemes, the tracks whose measurement indices on the past N scans are the same are regarded as "similar" and identified.

Thus they are often referred to as N-scan or depth-N methods. In Figure 3-4, hypotheses 1 and 2 can be combined if N = 3.

The justification is that since each track distribution is driven by the measurements assigned to it, if two tracks share the same measurements in the recent scans (data sets) they should be similar. This scheme is criticized by Reid [5] on the ground that some events in the past may have a greater influence than the most recent N scans. However, since the Markovian nature of a target model removes this possibility, the N-scan approach is attractive because of its simplicity.

After identifying tracks according to the N-scan or depth-N criterion, we may have several identical hypotheses, i.e., hypotheses with the identical set of tracks. Then those hypotheses are combined in a natural way. In a sense, this approach may be actually viewed as combining tracks rather than combining hypotheses. In fact, since similarity is initially defined at the track level even in distribution-oriented methods, one may further classify the combining techniques according to where combining takes place. For example, distribution-oriented combining may be performed at the track level or at a hypothesis level. While track-level combining may seem to be more straightforward, it creates the problem of how two distributions should be combined, since there is no natural weighting formula (similar to that of (3.25)) used in distribution-oriented combining at the hypothesis level.

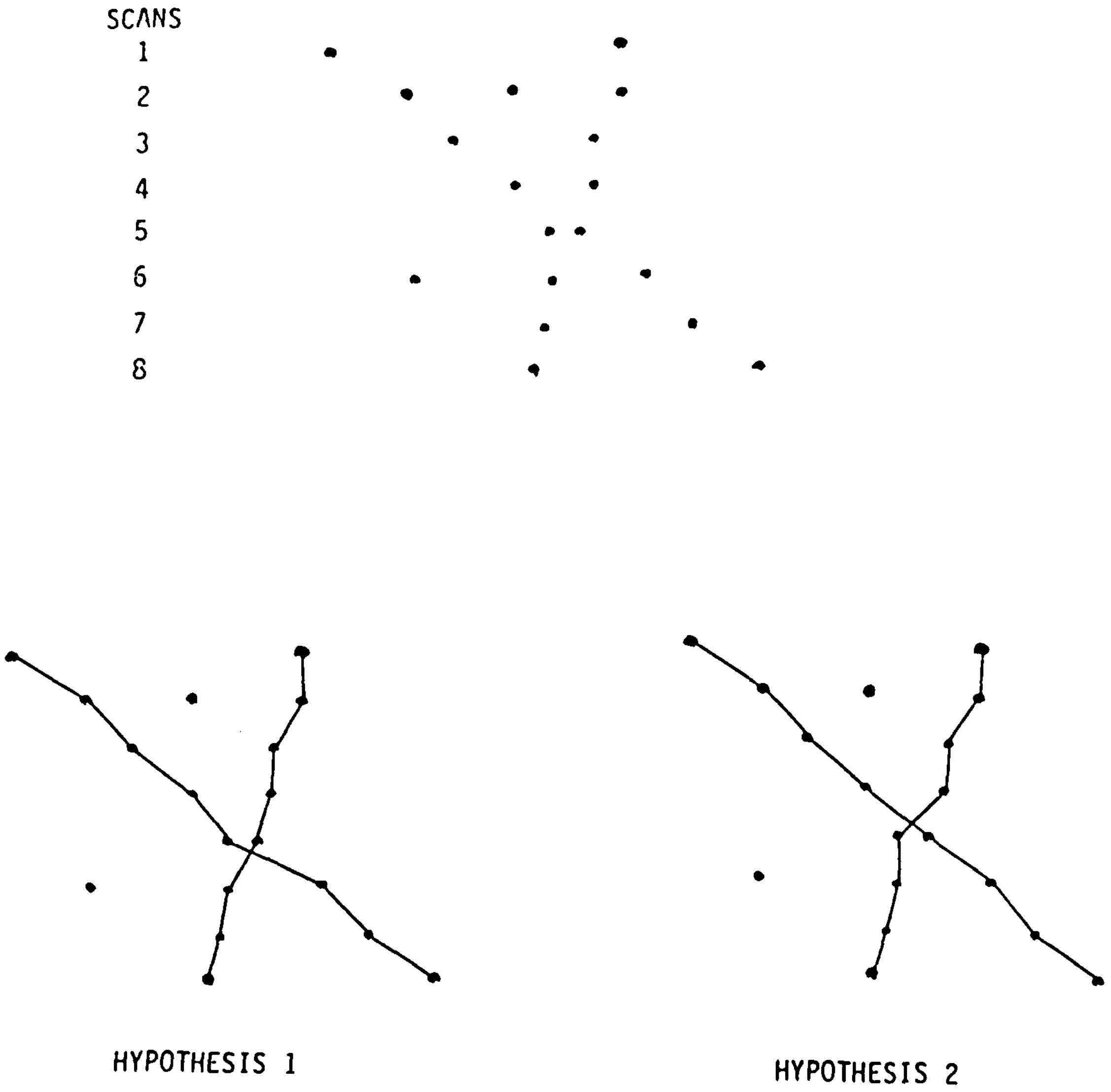

Figure 3-4: N-Scan Hypothesis Combining

To summarize, N-scan or depth-N methods have two major disadvantages, namely, the unresolved issues of (1) how to choose a right depth N, and (2) how to combine track state distributions. Just as in the case of pruning, many theoretical questions remain in combining hypotheses. Our current preference is distribution-oriented combining at the hypothesis level since there is a clear way for combining two probability distributions. The similarity condition, however, should be carefully chosen according to the physical nature of the particular problems and the chosen representation of each track distribution, etc.

3.5.3 Windowing

When i.i.d. target models are used, each measurement in a data set can be individually evaluated by likelihood functions. When a track state distribution has a reasonable variance and the measurement errors are not exceptionally large, one can expect the track-measurement likelihood to be very small for a measurement which is geometrically far from the expected position based on the state distribution associated with the track. Windowing techniques are generally designed to set an appropriate threshold so that the track-measurement likelihood in such a case becomes zero rather than a small positive number.

Thus, one may consider such techniques to be a special kind of pruning, i.e., immediate pruning of branches based solely on one likelihood function. In other words, windowing is a method for preventing all but a certain set of data from being even tentatively associated with each track. For this reason, such a process is often called data validation. When track state distributions and measurement error distributions are both Gaussian, windowing can be accomplished by a a classical chi-square test. As described in [12], this test may be performed in several steps. For example, the first step may consist of ℓ_∞ test (square test), and then a normalized square-of-innovation test (ellipse test), and finally, the likelihood function test itself.

Another view of windowing is that it is part of the distribution representation and modeling process. According to this view, when the track state and measurement distributions are modeled as Gaussian, they really are approximations of reality since such distributions can only have compact supports in the real world. For example, when the standard deviation of the measurement error is one mile, a data point 100 miles away from the mean of the track distribution should yield zero as its likelihood rather than a very small but positive number. We prefer this point of view to the pruning or approximation view. Thus any windowing process should be carefully designed to reflect the particular physical nature of the problem.

3.5.4 Clustering

The basic idea behind clustering is that two events taking place at locations far apart should be independent and can be evaluated separately. Clustering techniques have been described in algorithmic form in [5] for general cases and more rigorously in [8] for a special case. When adequate windowing is performed, there is a natural way to avoid redundant calculations in evaluating hypotheses since the posterior probability of each hypothesis is the product of an appropriate set of likelihood functions. This constitutes another view of clustering.

Mathematically, clustering can be defined as follows. Let $\mathcal{H}$ be the set of all non-zero-probability hypotheses at a given data set, i.e., for all $\lambda \in \mathcal{H}$,

$$p(\lambda | Z) > 0. \tag{3.26}$$

For each possible track τ, the posterior probability of τ is given by

$$p(\tau | Z) = \sum_{\tau \in \lambda \in \mathcal{H}} p(\lambda | Z). \tag{3.27}$$

Let $\mathcal{T}$ be the set of all non-zero-probability tracks, i.e., for all $\tau \in \mathcal{T}$,

$$p(\tau | Z) > 0. \tag{3.28}$$

Let $\mathcal{C}$ be any partition of $\mathcal{T}$ which satisfies the following condition:

For any pair (C', C'') of elements in $\mathcal{C}$, such that $C' \neq C$, and any $\tau' \in C'$ and $\tau'' \in C''$:

$$\tau' \cap \tau'' = \phi. \tag{3.29}$$

This condition means that the non-zero-probability tracks are partitioned so that no two tracks which are in different elements (clusters) of the partition share a common measurement index. For each C in $\mathcal{C}$, let:

$$\mathcal{H}_C = \{\lambda \cap C \mid \lambda \in \mathcal{H}\} \tag{3.30}$$

and

$$I_C = \cup C. \tag{3.31}$$

Then clustering is the process of generating $\{(I_C, \mathcal{H}_C) \mid C \in \mathcal{C}\}$. Each $\mathcal{H}_C$ is the set of local hypotheses which consists only of tracks in C. For each cluster $C \in \mathcal{C}$ and each local hypothesis λ_C in $\mathcal{H}_C$, define the local posterior probability p_C by

$$p_C(\lambda_C|Z) = \Sigma \, \{p(\lambda|Z) \mid \lambda \in \mathcal{H},\ \lambda \cap C = \lambda_C\}. \tag{3.32}$$

Then it is clear that

$$p(\lambda|Z) = \prod_{C \in \mathcal{C}} p_C(\lambda_C|Z) \tag{3.33}$$

Each global non-zero-probability hypothesis can thus be represented as a union of local hypotheses (one from each cluster) and its posterior probability is the product of the local probabilities of the local hypotheses. From this definition, we see that clustering involves partitions at all levels: hypotheses, tracks and measurements.

Equation (3.33) is called the orthogonality condition. According to the above definition of clustering, the orthogonality condition should hold whenever the non-intersection condition of the tracks given by (3.29) holds. However, when some approximation techniques such as pruning and combining are employed, orthogonality condition may not hold. The clustering technique described in [5] is a method in which the orthogonality condition is maintained without checking the non-intersection condition of the tracks. This technique can be described in terms of algorithmic procedures as follows:

1. Initialization of a Cluster - Whenever there is a measurement such that the newly detected target likelihood is not zero but the track-measurement likelihood with every existing track is zero, a new cluster should be created out of the measurement.

2. Cluster Merging - Whenever there is a measurement such that each of the corresponding track-measurement likelihood functions with two or more tracks in different clusters is non-zero (in other words, there is a measurement lying in the intersection of the validation regions of two tracks in two different clusters), such clusters should be merged before the measurement can be processed. The merging of the clusters is accomplished by forming the union of the tracks in the clusters, generating the global hypotheses and evaluating the global probabilities as the products of the local probabilities.

3. Cluster Splitting - Whenever a track with probability one, i.e., one contained in every local hypothesis in a cluster is found, it is split to form a new cluster consisting of one hypothesis with the sole track. Of course, the local probability of such hypothesis is one.

We believe that any serious attempt on practical problems should not be made without clustering, particularly in a problem involving a large geographic area. One may even assert that clustering is probably the most powerful hypothesis management technique in controlling the number of hypotheses. Of course, how successful a clustering technique is depends on the external conditions such as target density, measurement errors and target dynamics.

The clustering procedure described above does not necessarily guarantee the finest clustering. The finest clustering may be found according to our mathematical definition of clustering. The test of the non-intersection condition can be easily implemented if we identify two tracks with the same measurement indices in a certain number of the most recent scans just like in a measurement-index-oriented combining scheme. The orthogonality condition can be met by modifying the local probabilities using appropriate approximation techniques whenever possible. This constitutes a new cluster-splitting technique which may improve on that described above. Although this newly proposed version of clustering seems promising, we do not have any actual implementation experience yet.

4. INFORMATION FUSION IN DISTRIBUTED GENERALIZED TRACKER/CLASSIFIER

Each node in the DSN receives information from other nodes. This information has to be fused with the local information to obtain an improved assessment of the state of the world. In this section we shall discuss the fusion carried out at each node. While many ways of fusion are possible, our work is based on the following philosophy:

> Sufficient statistics for multitarget tracking are communicated through the network. Upon receiving these sufficient statistics, each node attempts to reconstruct the global sufficient statistics that would be available if the actual sensor data sets were communicated.

We shall present the distributed version of the Generalized Tracker/Classifier of Section 3. Our discussion in this section will be restricted to broadcast communication, where each node broadcasts its results to all the other nodes periodically. The derivations of these results, as well as their generalizations to more complex situations, are given in Appendix C.

4.1 PROBLEM STATEMENT

In this section we state the distributed multitarget tracking and classification problem.

4.1.1 Models

The target and sensor models are the same as those discussed in Section 3. Our emphasis is still on independent and identically distributed target models. The system now consists of a set of processing nodes called N. Each node n in the set processes the measurements from the set of sensors called S_n. We assume that the sensor sets for different nodes are disjoint, i.e., $S_n \cap S_{n'} = \phi$ for $n \neq n'$. On the other hand, the sensors of different nodes may have overlapping coverage.

In a general DSN, the nodes may communicate in many different ways. In this section, we restrict our attention to the broadcast type communication (more general communication for distributed estimation systems will be discussed in Section 5). The processing nodes communicate with each other at various times (the times need not be periodic and the communications need not be synchronized). When messages are broadcasted and received, each node in the network then updates its assessment on the state of the world.

We assume that between broadcast times, each node receives a large amount of data from the senors. Thus it is more efficient for the nodes to process the local data first before communication. In particular, each node broadcasts a set of possible hypotheses and the probability of each hypothesis, a set of possible tracks and the state distribution of each track, and the expected number of undetected targets based on the local information. These quantities from various nodes are then to be integrated or fused to obtain a better estimate. In order to define the problem properly, we need to generalize the definitions of tracks and hypotheses introduced in Section 3.

4.1.2 Tracks and Hypotheses

Let $\underline{K}$ be the data set index set as defined in Section 3, i.e., the set of all data set indices $k = (t,s)$. For each k in $\underline{K}$, the data set and the measurement index set are

$$z(k) = \left((y_j(k))_{j=1}^{N_M(k)}, N_M(k), k \right) \tag{4.1}$$

and

$$J_M(k) \times \{k\} = \{1,\ldots,N_M(k)\} \times \{k\}. \tag{4.2}$$

The set of all data sets is then

$$\underline{Z} = \bigcup_{k \in \underline{K}} z(k) \tag{4.3}$$

and the set of all measurement indices is

$$\underline{J} = \bigcup_{k \in \underline{K}} J_M(k) \times \{k\}. \tag{4.4}$$

Since each processing node generates hypotheses and tracks based on different information, we now generalize the definitions of tracks and hypotheses in Section 3 so that they can be defined on arbitrary subsets of $\underline{J}$. Let J be any subset of $\underline{J}$. Then a <u>track</u> on J is a (possibly empty) subset of J and a <u>possible</u> track is one which contains at most one measurement index in each single data set. A (data-to-data association) *hypothesis* on J is a (possibly empty) collection of tracks on J and a *possible* hypothesis is one containing only nonempty possible tracks such that no two distinct tracks intersect. Let $\mathcal{T}(J)$ and $\mathcal{H}(J)$ be the sets of possible tracks and possible hypotheses respectively.

Since there is a unique correspondence between the elements in the sets $\underline{J}$, $\underline{Z}$ and $\underline{K}$, we may call a track (or hypothesis) on $J \subseteq \underline{J}$ a track (or hypothesis) on K or Z if $K \subseteq \underline{K}$ such that

$$J = \bigcup_{k \in K} \{1,\ldots,N_M(k)\} \times \{k\} \tag{4.5}$$

and

$$Z = \bigcup_{k \in K} z(k). \tag{4.6}$$

For each subset Z of $\underline{Z}$, we can define an *information state* I(Z) consisting of the set of all possible tracks, the set of all possible hypotheses, the posterior probability of each hypothesis, the state distribution/density of each track, and the expected number of undetected targets, i.e.,

$$I(Z) = \{\mathcal{T}(Z), \mathcal{H}(Z), (p(\lambda|Z), \lambda \in \mathcal{H}(Z)), (p(x(t)|Z,\tau), \tau \in \mathcal{T}(Z)), \nu(Z) \tag{4.7}$$

where $p(x(t)|Z,\tau)$ and $\nu(Z)$ are as defined in Section 3. I(Z) is called the information state because it summarizes all the information about the targets contained in Z and is used in the recursion in the GTC.

4.1.3 Information Fusion Problem

The nodes are assumed to communicate in a broadcast mode. If the actual data sets were communicated, after one such communication time, all the agents would have the same information $\overline{Z}$ which is a subset of $\underline{Z}$. Let $I(\overline{Z})$ be the information state of $\overline{Z}$. Until the next communication time, each node n receives data sets from the sensors in S_n so that its information increases. The increasing information can be processed using the GTC. Let Z_n be the information of node n just before the next communication, and $I(Z_n)$ be the information state. If again the actual data sets were communicated, the information of each node would change to

$$Z = \bigcup_{n \in N} Z_n \tag{4.8}$$

with information state I(Z). Our assumption, however, is that the nodes only communicate their information states. Thus the problem is how to recover the information state I(Z) from $(I(Z_n))_{n \in N}$ and $I(\overline{Z})$. Figure 4-1 illustrates the structure of the problem.

There are two parts to this problem:

- Hypothesis Formation: Given $I(\overline{Z})$ and $(I(Z_n))_{n \in N}$, how should all

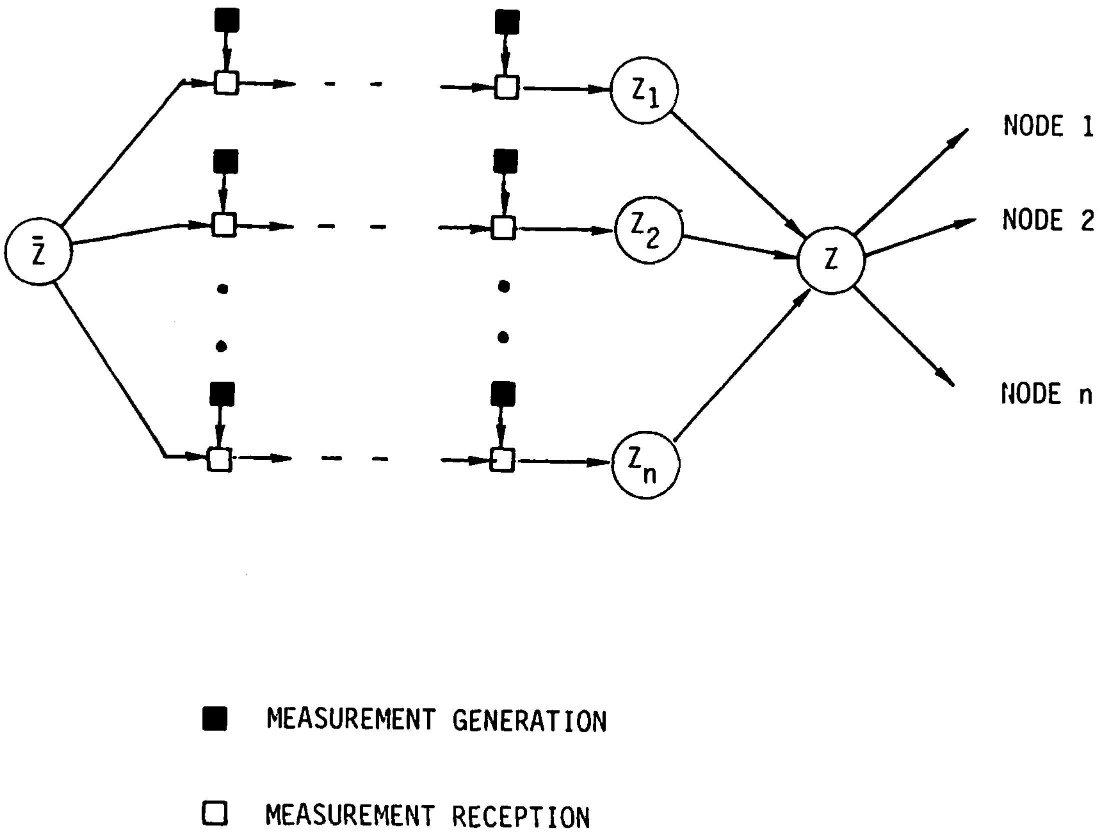

Figure 4-1: Structure of Information Fusion Problem

the possible tracks and hypotheses on Z be formed?

- Hypothesis Evaluation: Given $I(\bar{Z})$, $(I(Z_n))_{n \in N}$, and all the possible tracks and hypotheses on Z, how should the probabilities of the hypotheses and the state distributions of the tracks be computed to complete the description of I(Z)?

The information fusion module consists of three submodules which carry out these two functions of hypothesis formation and hypothesis evaluation and the additional function of hypothesis management. These will be discussed separately.

4.2 HYPOTHESIS FORMATION

The objective of this submodule is to generate all the possible hypotheses and tracks from the local hypotheses and tracks.

4.2.1 Example

The following example (Figure 4-2) shows that one has to be careful in forming the global hypotheses from the local hypotheses.

Consider two nodes each with one sensor. Node 1 has sensor 1, and node 2 has sensor 2. Sensor 1 (s=1) generates a data set with only one measurement at time 1 (t=1). This measurement can be indexed by (j,t,s) where j=1 indexes the only measurement within the measurement set. Node

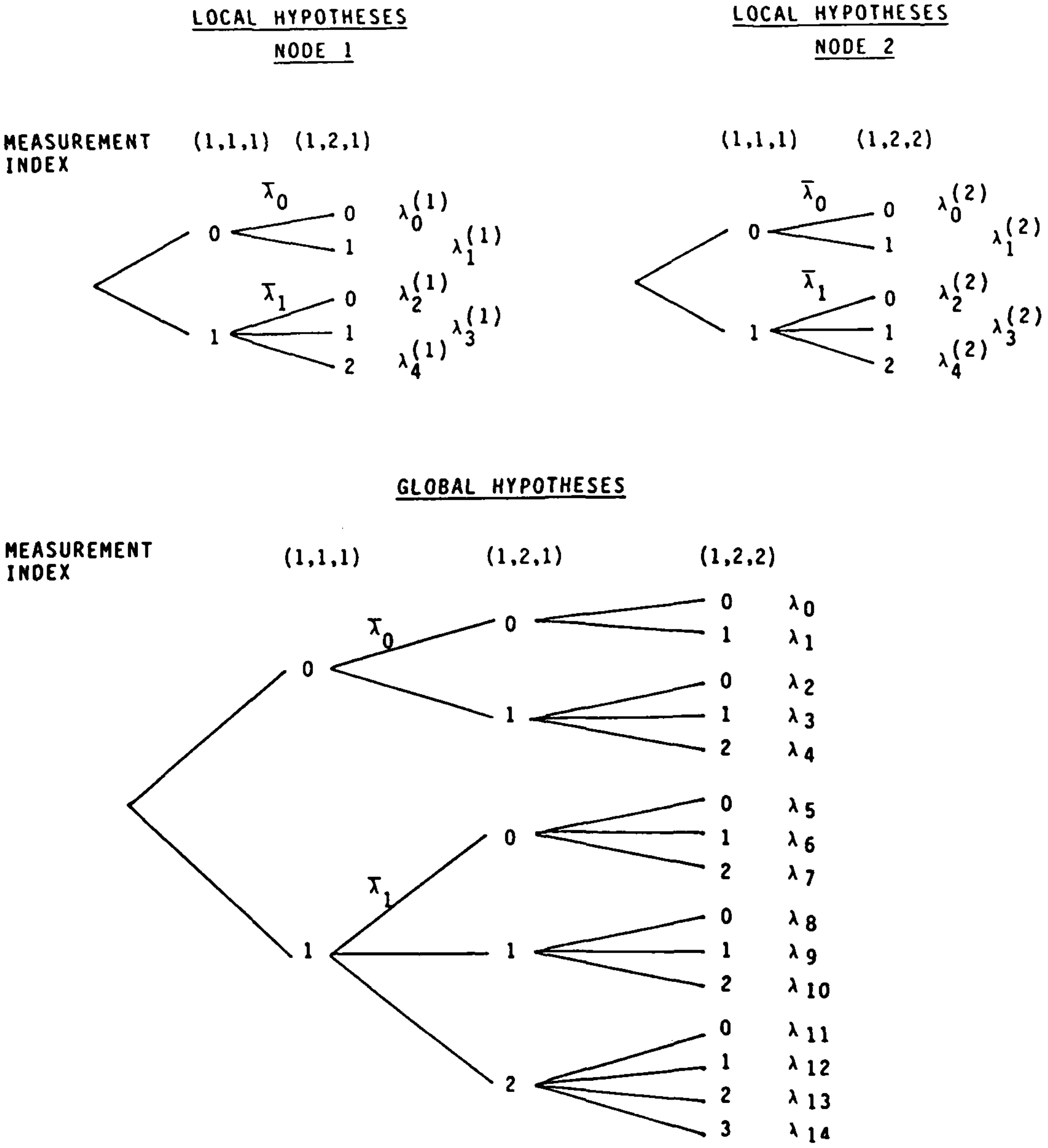

Figure 4-2: Local and Global Hypotheses - an Example

1 forms two hypotheses and broadcasts to node 2. Thus

$$\bar{J} = \{(1,1,1)\}. \tag{4.9}$$

At time t=2, each sensor generates another data set with one measurement each. These measurements can be indexed by (1,2,1) and (1,2,2) respectively. The new cumulative measurement index sets at the two nodes are then for n = 1, 2,

$$J_n = \{(1,2,n)\} \cup \bar{J} = \{(1,2,n),(1,1,1)\}. \tag{4.10}$$

Local hypotheses are formed for each node. The global cumulative measurement index set is

$$\begin{aligned} J &= J_1 \cup J_2 \\ &= \{(1,1,1),(1,2,1),(1,2,2)\}. \end{aligned} \tag{4.11}$$

Figure 4-2 shows the five local hypotheses defined on J_n, n = 1,2, and the fifteen (15) global hypotheses defined on J.

Note that if we compose the local hypotheses without any restriction, there would be twenty-five (25) hypotheses. This is in excess of the actual number of global hypotheses if all the data were processed in a centralized manner. Thus some of the compositions are inadmissible. On close examination, we discover that some compositions are inconsistent. For example, $\lambda_1^{(1)}$ from node 1 implies that (1,1,1) is a false alarm and $\lambda_2^{(2)}$ from node 2 implies that (1,1,1) is a target. These two obviously conflict and cannot be composed to form a global hypothesis. We also note that these two local hypotheses are expanded from different

$\bar{\lambda}$'s, i.e., they have different predecessors. This suggests that we should only compose pairs of hypotheses with common predecessors.

If we do that, the number of possible compositions is 13, which is less than the number of global hypotheses. This means that some compositions of local hypotheses should produce more than one global hypothesis. Table 4-1 shows how this happens. On the horizontal axis, we have local hypotheses for node 2; on the vertical axis, we have local hypotheses for node 1. Each entry in the matrix contains the global hypotheses composed from the pair of local hypotheses. An empty entry indicates an impossible composition. We note that two of the compositions yield more than one global hypothesis each. For example, both $\lambda_1^{(1)}$ and $\lambda_1^{(2)}$ have one track each. These two tracks can correspond to the same target or two different targets, thus resulting in two global hypotheses.

Table 4-1 Hypothesis Composition

		$\bar{\lambda}_0$		$\bar{\lambda}_1$		
		$\lambda_0^{(2)}$	$\lambda_1^{(2)}$	$\lambda_2^{(2)}$	$\lambda_3^{(2)}$	$\lambda_4^{(2)}$
$\bar{\lambda}_0$	$\lambda_0^{(1)}$	λ_0	λ_1			
	$\lambda_1^{(1)}$	λ_2	λ_3, λ_4			
$\bar{\lambda}_1$	$\lambda_2^{(1)}$			λ_5	λ_6	λ_7
	$\lambda_3^{(1)}$			λ_8	λ_9	λ_{10}
	$\lambda_4^{(1)}$			λ_{11}	λ_{12}	λ_{13}, λ_{14}

4.2.2 Hypothesis Formation Procedure

We have the following two level procedure for hypothesis formation:

1. **Hypothesis-to-hypothesis composition:**

   a. For each λ_n, identify the predecessor hypothesis $\lambda = \lambda_n | \bar{J}$, the restriction of λ_n to the past common cumulative measurement index set $\bar{J}$.

   b. Exhaust all possible compositions of local hypotheses with the same predecessor.

2. **Track-to-track composition:** For each collection of local hypotheses which can be composed,

   a. Construct a unique extension τ for every $\bar{\tau}$ in $\bar{\lambda}$ by letting $\tau = \bigcup_{n \in N} \tau_n$ where τ_n is the unique extension in λ_n of $\bar{\tau}$. Let λ_{OLD} be the set of such tracks.

   b. Exhaust all possible compositions of tracks which are not extensions of tracks in $\bar{\lambda}$. Let λ_{NEW} be one such set. Then any hypothesis composed from a given local hypothesis composition is of the form $\lambda_{OLD} \cup \lambda_{NEW}$.

For the example in Figure 4-2, the operations of Level 1 generate altogether 13 possible hypothesis-to-hypothesis compositions with 4 from the predecessor $\bar{\lambda}_o$ and 9 from the predecessor $\bar{\lambda}_1$. Except for the compositions of $(\lambda_1^{(1)}, \lambda_1^{(2)})$ and $(\lambda_4^{(1)}, \lambda_4^{(2)})$, λ_{NEW} is unique, and thus there is only a single global hypothesis for each composition. For the pair $(\lambda_1^{(1)}, \lambda_1^{(2)})$, $\lambda_{OLD} = \{\phi\}$ and there are two λ_{NEW}'s, resulting in two global hypotheses. For the pair $\lambda_4^{(1)}$ and $\lambda_4^{(2)}$, $\lambda_{OLD} = \{(1,1,1)\}$ and there are again two λ_{NEW}'s, resulting in two global hypotheses.

4.3 HYPOTHESIS EVALUATION

Given the global hypotheses and global tracks constructed from the local hypotheses and tracks, the objective of the hypothesis evaluation submodule is to compute their probabilities and state distributions using the communicated local information. Specifically, we need to evaluate $p(\lambda|Z)$ for all global hypotheses, $p(x(t)|Z,\tau)$ for all global tracks and $\nu(K)$, the expected number of undetected targets.

4.3.1 Static Target Models

We first state the hypothesis evaluation algorithm for static target models. Since deterministic random process models can be reduced to static models, this algorithm is also useful when the targets can be approximated by deterministic random processes, e.g., when the driving noise in a linear stochastic system is very small. The following results are derived in Appendix C.

Fusion Algorithm

For stationary targets and broadcast communication, we have for every $\lambda \in H(J)$,

$$p(\lambda|Z) = \tilde{C}^{-1} p(\bar{\lambda}|\bar{Z})^{-(\#N-1)} \Big(\prod_{n \in N} p(\lambda_n|Z_n)\Big) \prod_{\tau \in \lambda} \ell_\tau \tag{4.12}$$

where $\tilde{C}$ is a normalization constant, $\#N$ is the number of elements in N, and

$$\ell_\tau = \int \frac{\prod_{n \in N} \tilde{p}(x|Z_n,\tau)}{(\tilde{p}(x|\bar{Z},\tau))^{\#N-1}} \; \mu(dx). \tag{4.13}$$

The expected number of targets undetected up to K is:

$$\nu(K) = \int \frac{\prod_{n \in N} \tilde{p}(x|Z_n,\phi)}{(\tilde{p}(x|\bar{Z},\phi))^{\#N-1}} \; \mu(dx) \tag{4.14}$$

$\tilde{p}(x|Z_n,\tau)$ and $\tilde{p}(x|\bar{Z},\tau)$ are given by

$$\tilde{p}(x|Z_n,\tau) = \begin{cases} p(x|Z_n,\tau) & \text{if } \tau \neq \phi \\ \nu(K_n)p(x|Z_n,\phi) & \text{if } \tau = \phi \end{cases} \tag{4.15}$$

$$\tilde{p}(x|\overline{Z},\tau) = \begin{cases} p(x|\overline{Z},\tau) & \text{if } \tau \neq \phi \\ \nu(\overline{K})p(x|\overline{Z},\phi) & \text{if } \tau = \phi \end{cases} \tag{4.16}$$

$p(x|Z_n,\tau)$ and $p(x|\bar{Z},\tau)$ are the state distributions at the time of fusion conditioned by track τ, Z_n and $\bar{Z}$. Furthermore, the state distributions can be fused to obtain

$$p(x|Z,\tau) = \frac{\prod\limits_{n \in N} p(x|Z_n,\tau)}{(p(x|\overline{Z},\tau))^{\#N-1}} \; C^{-1} \tag{4.17}$$

where C is a normalization constant.

Similar to hypothesis formation, this fusion formula (4.12) has again a two level structure. At the high level, we consider the probabilities of the local hypotheses. The probability of the global hypothesis is a product of the probabilities of the local hypotheses. However, because each of the local probabilities has been computed using the prior probability of the predecessor hypothesis, the product has to be divided by the (#N-1)-th power of the probability of the common predecessor hypothesis to prevent any double counting of this probability. This elimination is quite standard in distributed estimation problems. When the information from multiple dependent sources is to be combined or fused, the redundant information has to be removed. A more complete discussion of this will be found in the next section and in Appendix B.

Given each hypothesis-to-hypothesis composition, the lower level considers the likelihood of each track. Specifically, ℓ_τ evaluates the likelihood that the multiple local tracks correspond to a single global track. ℓ_τ is related to an integral of the product of the state distributions of the local tracks. Thus if the local tracks have very similar statistics, e.g., position means and variances, the product will be a non-zero function and the integral will be large, resulting in a high likelihood for the track. On the other hand, if the state distributions of the local tracks are very different, they will have little overlap and the integral will be small or zero. In this case, the likelihood that the two local tracks correspond to the same target will be small. The division by the state distributions of the common prior tracks is again to prevent double counting of any common information.

The updating of the expected number of undetected targets and the state distributions of the global tracks have similar equations. In the case of Gaussian distributions, the computation only involves means and covariances.

4.3.2 Dynamic Target Models

If the targets are dynamic, as in more realistic situations, the hypothesis evaluation formula has the same form as in the static case. However, the track likelihood functions must now be computed differently. All the conditional probabilities of x must now be replaced by those of x_I, where x_I is the target state evaluated at the set

$$T_I = \{t \mid (t,s) \in K-\bar{K}\}, \tag{4.18}$$

the times when sensor observations are made since the last communication time. In other words, to decide whether two local tracks could have come from the same target, we consider the probability distributions of the state for these two tracks over the entire time interval.

For a special class of random processes, called deterministic processes, where the state at any time uniquely determines the processes, the track likelihoods depend only on the state distributions of the tracks at $\min(T_I)$. Figure 4-3 illustrates how the likelihoods are computed for the three different processes. For both static and deterministic random processes the densities of the target states at a single time are needed in track likelihood computation. For a general random process, the densities of the states over one time interval are needed. This, of course, makes hypothesis evaluation more difficult. In many situations, however, such processes can be approximated by deterministic random processes. This is the case when the noise driving the linear system which models the target motion is very small.

4.4 HYPOTHESIS MANAGEMENT

Hypothesis management is again needed to keep the number of hypotheses manageable within the computational resources of each node. The same hypothesis management techniques discussed in Section 3.5 are again applicable.

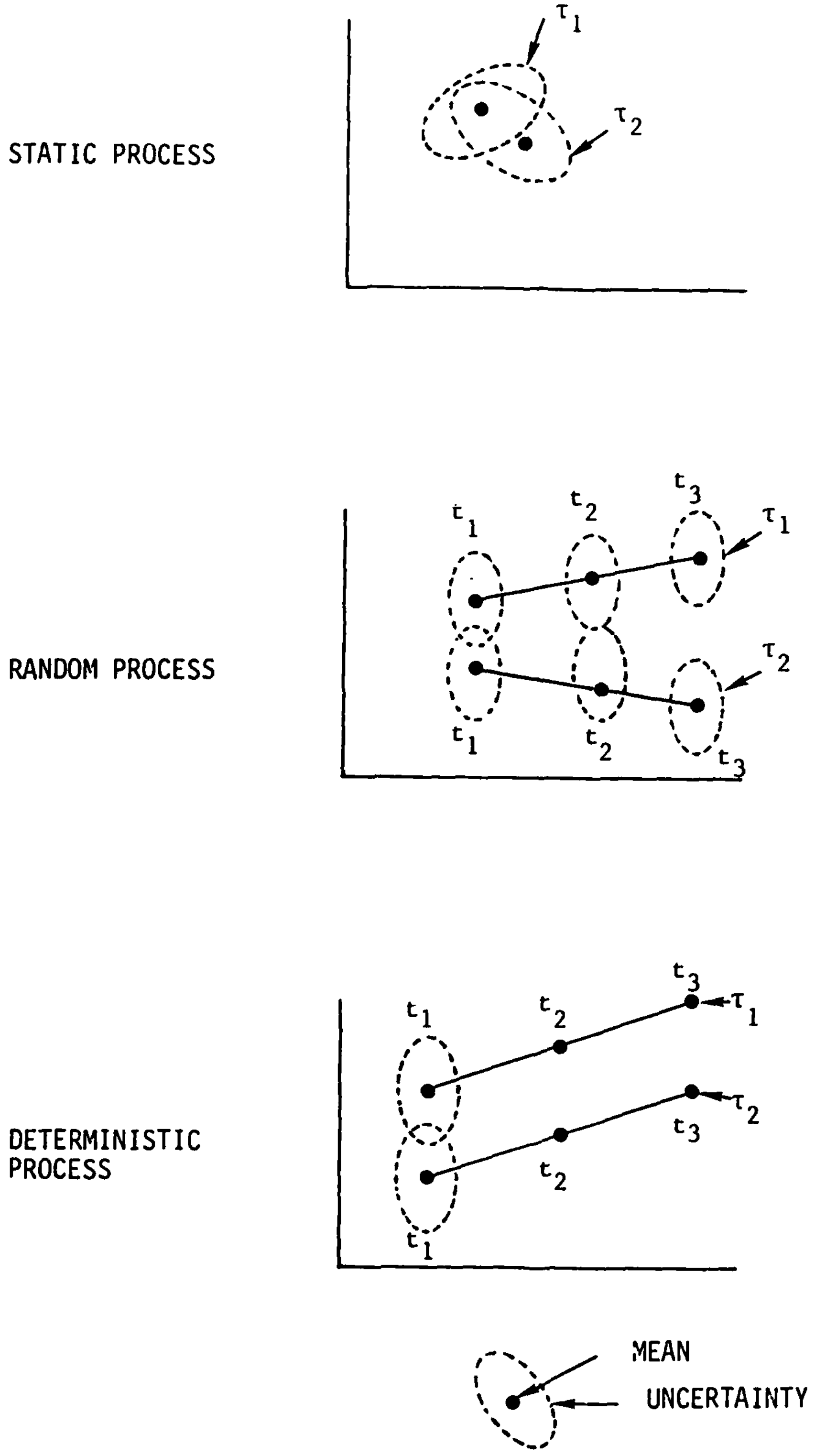

Figure 4- 3: Computation of Track Likelihoods

5. INFORMATION DISTRIBUTION AND PROBLEMS IN DISTRIBUTED ESTIMATION

The information distribution module in each node distributes the local information to the other nodes in the network. It thus determines the following:

- when to send a message
- where to send the message
- what to include in the message

The three items above together constitute the information structure of the system. The design of information structures is one of the most difficult problems related to the DSN since it is not amenable to analytic studies. However, an efficient design can serve the objective of getting the needed information to the right node at the right time without using too much communi:ation resources. In this project, we have studied these issues through simulation experiments.

In Section 4, we have discussed the information fusion problem for distributed multitarget tracking assuming a broadcast type communication. A key feature of the information fusion algorithm is that any redundant information contained in dependent data sets has to be removed. If the communication pattern is different from the broadcast type, information fusion becomes more complicated. Since not much has been done in this area, our discussion will be restricted to general estimation problems rather than multitarget tracking. A more detailed description can be found in Appendix B and [4].

5.1 WHAT TO DISTRIBUTE

We adopt the philosophy that each node, upon receiving messages from other nodes, attempts to reconstruct the best assessment of the state of the world as if the actual sensor reports were communicated. Thus the messages should represent the information state at each node. In the context of multitarget tracking and classification, this information state corresponds to the set of hypotheses, the set of tracks, the probabilities of the hypotheses, the state distributions of the tracks and the expected number of undetected targets. When all of these are communicated, as discussed in the previous section, each node is then able to reconstruct the global hypotheses and their probabilities.

If communication constraints are present, the information to be distributed to the other nodes may have to be reduced. In this case, only a subset of the hypotheses may be distributed. The information fusion algorithm described in the previous section will no longer be optimal. The question is then: how many hypotheses ought to be kept. This problem is similar to the hypothesis management problem discussed in Section 3. Again, no general theory is available. Rather, communication is dictated by practical considerations such as the bandwidth of the network. Frequently, only the best hypothesis from each node can be sent.

A more adaptive strategy is to vary the communication based on the information present in the set of hypotheses. For example, one adaptive strategy is to transmit a sufficient number of hypotheses until the cumulative probability exceeds a certain prescribed value. This is similar to the adaptive pruning strategy described in Section 3.5.1. If one hypothesis stands out as being highly probable, then only that hypothesis will be distributed. On the other hand, if several hypotheses are equally probable, then they should all be distributed. In some situations, it may be desirable to go one step further. The information distributed should not only be decided by the transmitting node, but depend on requests from potential receivers. Thus a node which is highly confused may request a lot of information from other nodes to help to disambiguate the situation. Although we have not implemented these adaptive strategies in our current system, they should be included in an improved version.

5.2 ISSUES RELATED TO DISTRIBUTED ESTIMATION NETWORKS

A DSN is a special case of a general distributed estimation network. A main advantage of such systems is that there is no single central node whose failure or destruction may disable the entire system. For such a distributed system to be really effective, the communication network supporting the nodes should have a certain amount of redundancy. Otherwise, some nodes may be isolated from others in a failure. A redundant network, however, means that the messages arriving at a node by different paths may contain the same information. If this redundant information is not removed in the processing, the same information may be used multiple times, resulting in an incorrect assessment of the state of the world.

Appendix B contains a theory for handling the distributed estimation problem for arbitrary network structures. A distributed fusion formula for combining the local conditional probabilities of the state of a system to obtain the optimal global conditional probability is given. This theory, together with the theory of multitarget tracking described in Section 3, is then used to develop a theory for distributed multitarget tracking and classification. In the following, we outline this theory and illustrate the importance of proper processing with an example.

5.2.1 Basic Results

Let x be the state or random variable to be estimated. Let S be a set of sensors. At a given time t in T, a sensor s generates an output or measurement z(t,s). Let $\underline{Z}$ be the set of all such measurements. Assume that the measurements are all conditionally independent given the state x.

Let N be the set of processing nodes or estimation agents, each receiving the reports from a subset of S. The information available to an agent at any time is a subset of $\underline{Z}$. We assume that each agent n at time t computes the conditional probability of x given the available information denoted by Z(t,n). Agents also communicate the conditional probabilities to other agents. When a conditional probability is received by an agent, it fuses this with its local conditional probability to obtain the conditional probability which would have resulted if the actual measurements, instead of the conditional probabilities, had been communicated through the network.

To carry out this fusion properly, the redundant information contained in the conditional probabilities has to be removed. The following lemma contains the basic results.

<u>Lemma</u>: Let Z_1, Z_2 be two subsets of $\underline{Z}$. Then

$$p(x|Z_1 \cup Z_2) = C \frac{p(x|Z_1)p(x|Z_2)}{p(x|Z_1 \cap Z_2)} \tag{5.1}$$

where C is a normalization constant.

This lemma can be regarded as a distributed version of Baye´s rule and is crucial in distributed estimation problems. Its proof can be found in Appendix B. The set $Z_1 \cup Z_2$ is the joint information in Z_1 and Z_2 while the set $Z_1 \cap Z_2$ is the common information. In combining the conditional probabilities $p(x|Z_1)$ and $p(x|Z_2)$ to obtain $p(x|Z_1 \cup Z_2)$, we note that both of these local probabilities have used the common information. Thus, if we combine these probabilities with a naive rule, e.g., by a product rule, the common information represented by $p(x|Z_1 \cap Z_2)$ would have been used twice. The redundancy can be removed by dividing with $p(x|Z_1 \cap Z_2)$. In other words, the lemma states that the probabilities $\{p(x|Z_1),p(x|Z_2)\}$ do not contain enough information to recover $p(x|Z_1 \cup Z_2)$ and that an additional probability $p(x|Z_1 \cap Z_2)$ is also needed.

This lemma can be used as a basis for developing optimal fusion formulas for arbitrary information structures. In a general case, each node would have to know the history of the messages and remember some statistics from the past.

5.2.2 Example

The following simplified example illustrates the removal of redundant information in an arbitrary network. Consider a network with three nodes {1, 2, 3}. Node i gets the measurements from sensor i. The state x is a discrete random variable with three possible values {a, b, c}. The a priori probability of x is uniform, i.e.,

$$p(x=a) = p(x=b) = p(x=c) = \frac{1}{3} \tag{5.2}$$

The sensor measurement z(t,s) also takes values in {a, b, c}, and the conditional probabilities of the measurements for all the sensors are given in Table 5-1.

Table 5-1 Sensor Characteristics

$p(z(t,1)|x) = p(z(t,3)|x)$

x z	a	b	c
a	0.4	0.3	0.3
b	0.3	0.4	0.3
c	0.3	0.3	0.4

$p(z(t,2)|x)$

x z	a	b	c
a	0.6	0.2	0.2
b	0.2	0.6	0.2
c	0.2	0.2	0.6

At each time $t = 0, 1, \ldots,$ each node processes the sensor data and computes the conditional probabilities of x given all the available information. At a time $s = t + d$, where d is a small time interval, each node sends the conditional probability to its immediate neighbor according to the graph of Figure 5-1. The communication is cyclic and counter-clockwise. Note that the information sent from one node eventually returns to the same node. Upon receiving the conditional probabilities, each node combines them to improve on the local estimate. Three algorithms for information fusion are considered.

- **Optimal Algorithm**: Let $Z(t,i)$ be the data available to a node i at time t if the actual measurements were communicated through the network instead of the conditional probabilities. Then the optimal fusion algorithm is

$$p(x|Z(s,i)) = C \frac{p(x|Z(t,i))}{p(x|Z(t-2,i))} \frac{p(x|Z(t,[i+1]))}{p(x|Z(t-1,[i+1]))} p(x|Z(s-3,i)) \qquad (5.3)$$

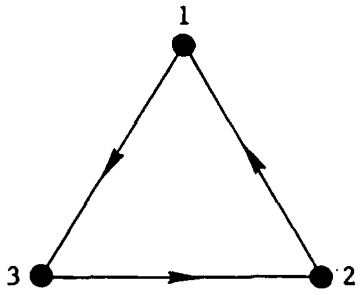

Figure 5-1: Communication Structures

where C is a normalization constant and [i] is i modulo 3. In this algorithm, in addition to the local conditional probability p(x|Z(t,i)) and p(x|Z(t,[i+1])) from the neighbor, each node has to remember some of the earlier probabilities in order to remove the redundant information. In addition, the history of the message is also needed.

- **Heuristic Algorithm 1**: In this algorithm, we assume that the conditional probabilities from the nodes do not contain any redundant information. Thus the fusion algorithm is given as a product

$$p(x|Z(s,i)) \approx C\ p(x|Z(t,i))p(x|Z(t,[i+1])) \qquad (5.4)$$

where C is again a normalization constant.

- **Heuristic Algorithm 2**: Heuristic algorithm 2 is similar to 1 in that it assumes to redundant information. However, it includes explicitly the suboptimal nature of the algorithm and tries to remain more objective. This is accomplished by having

$$p(x|Z(s,i)) \approx C\ [p(x|Z(t,i))p(x|Z(t,[i+1]))]^{0.5} \tag{5.5}$$

Since the square root operation has the effect of flattening out a probability distribution, this algorithm is less likely to have complete confidence in any particular conclusion and is more willing to incorporate new information. Thus it may be viewed as a hedging strategy.

The results of the simulation studies are shown in Table 5-2. The optimal algorithm and heuristic algorithm 2 both converge to the true value of x, although the convergence of heuristic algorithm 2 is slower. Heuristic algorithm 1 converges very fast but frequently to the wrong value of x. In terms of memory requirements, the two heuristic algorithms are quite similar while the optimal algorithm requires more memory.

Table 5-2 Simulation Results

Algorithm	Percentage of Correct Classifications	Memory Requirements	Convergence Speed
Optimal	100	High	Fast
Heuristic 1	60	Low	Very Fast
Heuristic 2	100	Low	Medium

Figure 5-2 shows a typical simulation run where heuristic algorithm 1 converges to the wrong value b while the true state is a. The conditional probabilities at each node are displayed for the states x = a and x = b. We can explain the behavior of heuristic algorithm 1 as follows. At time t = 1, sensor 2 gets an erroneous measurement b. Since sensor 2 is assumed to be quite accurate, the local conditional probability is biased towards x = b. Node 1 incorporates this into its local probability at time 2, resulting in a high value for the conditional probability of x = b. This bias is propagated to node 3 at time 3. In the meantime, node 2 has obtained some correct measurements. However, the communication from node 3 arrives and increases its confidence in x = b. From then on, the process gets out of control and the conditional probabilities all converge rapidly to x = b.

If we observe the behavior of heuristic algorithm 2, we notice that it never concludes that the state is any value with probability 1. With this kind of hedging, the algorithm is more likely to recover after an error has been made.

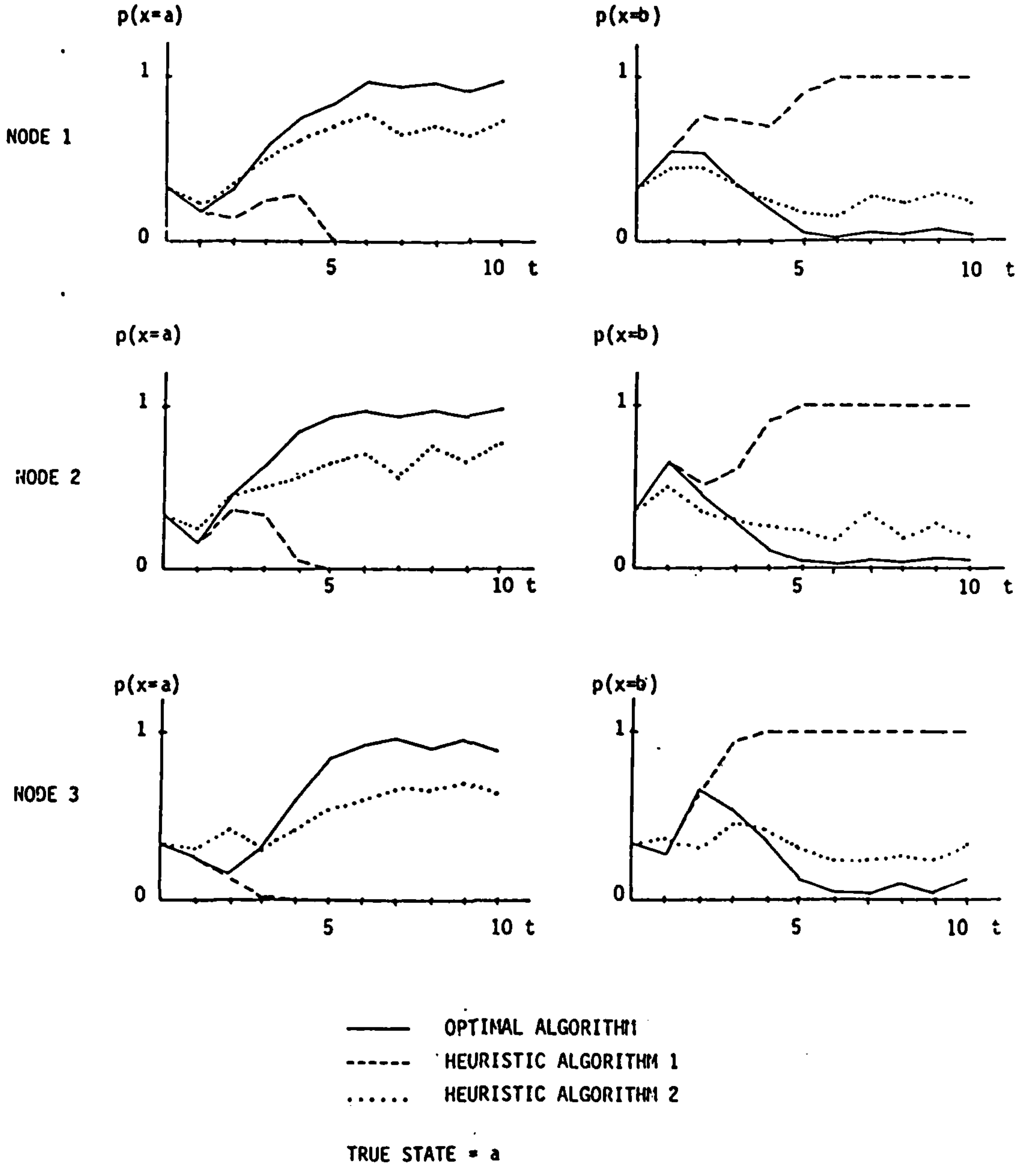

Figure 5-2: A Typical Simulation Run

6. NUMERICAL EXAMPLES

6.1 INTRODUCTION

In this section, two numerical examples are described to illustrate the performance of the distributed Generalized Tracker/Classifier (GTC) and to compare its performance with those of alternative structures. For both examples, simple target dynamics were chosen; in fact, the first example assumes stationary targets. Simple dynamics were chosen for the following reasons. First, target dynamics affects mostly the filtering problem, which, although is always at the bottom of the GTC or distributed GTC algorithm, is not the main focus of our research. Second, if complex dynamics are to be used, many numerical approximations, sometimes very bold ones, must be employed, making it difficult to single out the major factors affecting the performance. Third, although as discussed in earlier sections, general target dynamics can be treated at least in principle, meaningful performance analysis for complicated dynamics requires further improvements in the efficiency of the GTC and distributed GTC computer codes. For the same reasons, very simple sensor models were chosen.

The underlying basic assumptions are:

1. Targets are distributed along a straight line (e.g., on a road or

in an air corridor).

2. Targets are either stationary (the first example) or moving at almost constant velocities (the second example).

3. Sensors measure position and velocity along the road. These models roughly approximate range-range rate radars whose angles are very narrow, and which point to the line segment at very shallow angles, so that the range readings can be regarded as a linear function of the 1-dimensional position corrupted by some noise.

4. In the first example, it is assumed that either the target speeds are very small compared with sensor revisit times or sensors are tuned to detect stationary targets.

5. In the second example, it is also assumed that the sensor readings include noise-corrupted linear measurements of the target positions and velocities.

Furthermore, we also assume the independent and identically distributed target models for which the current version of GTC and distributed-GTC have been designed.

6.2 STATIONARY TARGET EXAMPLE

We first consider the case of stationary targets. This can be used to approximate targets whose movements are small within the observation interval.

6.2.1 Example Scenario

In this example, each individual target is represented by a one-dimensional position and is assumed to be stationary. Therefore, the state of each target is a real number. The field-of-view of each sensor is assumed to be identical and to be the line segment [0,L]. Each sensor creates a measurement of the form

$$y = x + \text{noise} \tag{6.1}$$

when it detects a target at x. The noise is an independent Gaussian random variable with standard variation σ which is small compared to L. The probability of detecting a target at x is given by

$$p_D(x) = \frac{p_{D\ \max}}{\sqrt{2\Pi}\ \sigma} \int_0^L e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2} dy \tag{6.2}$$

if $x \in [-3\sigma, L+3\sigma]$ and 0 otherwise. The probability density function $p_m(.|x)$ of a measurement y given that it originates from a detected target x is given by:

$$p_m(y|x) = \frac{e^{-\frac{1}{2}\left(\frac{y-x}{\sigma}\right)^2}}{\int_0^L e^{-\frac{1}{2}\left(\frac{\eta-x}{\sigma}\right)^2} d\eta} \tag{6.3}$$

if $y \in [0, L]$ and zero otherwise. It is assumed that all the sensors are modeled identically.

The target positions are independent and identically distributed and the common distribution is uniform on $[-3\sigma, L+3\sigma]$. The total number of targets is constant but random and unknown; its a priori distribution is Poisson with mean ν_T. The target density is then

$$\beta_T = \nu_T/(L+6\sigma) \tag{6.4}$$

The number of false alarms in each scan is also Poisson with mean ν_{FA} for each sensor. The adaptive-breadth/adaptive-thresholding pruning technique described in Section 3 is employed for the GTC and the distributed GTC. The threshold level is represented by $\varepsilon \in (0, 1)$.

6.2.2 Communication Schemes and Performance Measures

Three communication schemes are examined:

- [Scheme 1] decentralized ---- single sensor (15 scans),
- [Scheme 2] centralized ---- two sensors (15 + 15 scans),
- [Scheme 3] distributed ---- two single-sensor nodes with communication/information fusion every 5 scans.

In the distributed scheme, all the hypotheses at each node are communicated. The centralized scheme can be regarded as the case when the sensor data are communicated every scan. Since we assume identical sensors with identical fields-of-view and with the same performance, the difference between Scheme 1 and Scheme 2 is only in the number of sensors or equivalently the number of scans in a centralized GTC system.

For each scheme the baseline parameters are given in Table 6-1.

Table 6-1 Baseline Parameters for Stationary Case

Pruning Threshold	ε	.05
Target Density	β_T	5/L
Measurement Error	σ/L	.001
Expected Number of FA	ν_{FA}	5/scan
Probability of Detection	P_{Dmax}	.7

To determine the sensitivity of the schemes to parameter variations each parameter in the above table was varied and used in Monte Carlo simulations which examined the performance of each scheme.

The performance is measured by several indices which are shown in the following table:

Table 6-2 Performance Indices

NAME	SYMBOL	MEASURE
Average Number of False Tracks	N_{FT}	False Alarm Rate of GTC Systems
Average Number of Missed Targets	N_{MT}	False Dismissal Rate of GTC Systems
Probability of Perfect Association	P_{PA}	Overall Performance Criterion of GTC Systems
Position Estimate Error	σ_{pos}	Accuracy of Position Estimates Given Correct Association
Average Execution Time	T_E	Actual CPU Time Consumed by GTC Systems Per Scan
Average Number of Tracks	N_{TR}	Number of Tracks Stored in GTC Systems Per Scan

At the end of each simulation run, the best hypothesis is examined and compared with the true positions of targets according to the following procedure. A pair (i, τ) of a target i and a track τ in the best hypothesis is called permissible if $|x - \hat{x}| < \varepsilon_E$ where x is the position of the target, $\hat{x}$ is the best position estimate base on the track τ and ε_E is the evaluation threshold. Then a maximum number of independent (non-overlapping) permissible pairs is obtained by running a modified, rectangular, threshold Munkres algorithm [13]. The result yields a set of correctly associated pairs. A track τ in λ is said to be correctly associated with a target i if (i, τ) is a correctly associated pair. Otherwise it is declared as a false track. A target i is said to be missed if there is no τ in λ such that (i, τ) is a correctly associated pair. Perfect association is by definition the case when there is no false track and missed target. The first three performance indices, N_{FT}, N_{MT} and P_{PA}, are thus calculated. The position error is then averaged over all the correctly associated pairs to yield σ_{pos}. The last two indices are rough measures for the computational requirements. T_E is measured by the average CPU time used to complete one entire Monte Carlo run. N_{TR} is the average number of tracks processed at each moment.

6.2.3 Experimental Results

Because of the stationary targets and the simple sensor model, the filtering required is very simple and, in most cases, reduces to one of the simplest forms of Kalman filtering. Whenever necessary, such as when a track state distribution is centered at the edge of the sensor's field-of-view and should be updated as a missed measurement, an appropriate quantization approximation is used to perform filtering or to calculate the track-measurement likelihood functions. A quantization approximation is also used to calculate the distribution of undetected targets.

Table 6-3 Baseline Quantitative Comparison

	P_{PA} Prob. of Perfect Association	N_{FT} Average No. of False Tracks	N_{MT} Average No. of Missed Targets	σ_{pos} Position Estimation Error (*)	T_E Average Execution Time (**)	N_{TR} Average No. of Tracks
Decentralized	.62	0	.54	1.19	2.69	7.32
Centralized	.70	.02	.32	.238	4.29	5.27
Distributed	.78	.04	.24	.403	5.95	6.01

(*) Normalized by the standard deviation (= .001 x field of view)

(**) Average execution time for one run:

Decentralized:	15 Data Sets
Centralized:	30 Data Sets
Distributed:	30 Data Sets + 3 Fusion Operations

Table 6-3 shows the results of a 50-run Monte Carlo simulation. The table compares the performance of the three schemes using the baseline parameters shown in Table 6-1. For each scheme we have assumed that every statistical parameter is known exactly and is used to calculate the track-measurement likelihood functions. As far as the first three criteria (which represent the target detection capability of the

GTC or the distributed GTC) are concerned, the performance of the centralized (Scheme 2) and the distributed (Scheme 3) schemes are almost identical. Since the expected number of true targets is 5, both N_{FA} and N_{MT} are very small. Thus both schemes perform very well and screen out the false alarms, five of which appear on the average in every scan. This similarity in performance is not surprising since the information fusion algorithm used in the distributed scheme is designed so that the results of the centralized scheme may be reconstructed from the partial information of the nodes. The performance of the distributed scheme is, however, slightly better than that of the centralized case, indicating the advantage of distributed calculation when hypothesis pruning is employed (pruning threshold = .02). On the other hand, the performance of the decentralized scheme (Scheme 1) shows how poorly the system performs when the amount of data is half that of the other schemes. Table 6-3 indicates an apparent performance degradation due to the the relatively small amount of data. However, this degradation is highly dependent on the quality of data (false alarm rate), the pruning threshold, etc., and will be discussed later.

Because of the relatively low probability of detection ($p_{Dmax} = .7$) and the hypothesis combining used in both the centralized GTC and the distributed GTC, the position error σ_{pos} is much larger than that from filtering with known origins of all the measurements. In particular, in the decentralized scheme with small amount of data, σ_{pos} exceeds even the sensor measurement error level. The difference between the centralized and distributed schemes seems to result from the pruning and combining in the information fusion program used in the distributed scheme. The last two criteria, T_E and N_{TR}, roughly reflect the computational time and space requirements of each scheme. The number of hypotheses and tracks increases rapidly as the amount of data increases. This rapid growth is, however, controlled by the pruning and combining of hypotheses. As seen in Table 6-3, the resultant increase in the CPU time is about 60% compared with 100% increase of data. The distributed scheme uses up about 40% more CPU time than the centralized scheme. When we consider the fact that the distributed scheme maintains two copies of data and processes them separately and that a relatively complicated fusion program is used, this increase seems very moderate.

When hypotheses are evaluated successively as every new data set arrives (as in the current implementation), the uncertainty in the origin of each measurement is resolved rapidly after enough data is accumulated. With less data available, the confusion may not be resolved to the end of one run, and thus many hypotheses (and accordingly many tracks) must be stored. The difference of N_{TR} between the decentralized and the centralized schemes reflects this effect correctly. The distributed scheme needs to store slightly more tracks than the centralized scheme. However, when we consider the fact that each node has its own copy of the tracking data (significant redundancy), this difference is surprisingly small. The qualitative comparison of the three schemes is summarized in Table 6-4. It should be noted, however, that the space requirement comparison does not include the size of the program itself.

Table 6-4 Baseline Qualitative Comparison

Criteria / Scheme	Detection (Tracking)	State Estimation	CPU Time Requirement	(*) Space Requirement
Decentralized	Fair	Poor	Low	Large
Centralized	Good	Good	High	Small
Distributed	Good	Fair	High	Small

(*) Excluding the program space.
The same program is used both for the decentralized and the centralized schemes.
The distributed scheme's program size is at least twice as large.

Figures 6-1 to 6-5 show the comparative statistics and are obtained by varying the base parameters shown in Table 6-1. Each point has been created by a 50-run Monte Carlo simulation. For each run, we have assumed that all the a priori statistics are exactly known and the track-measurement likelihood functions are calculated accordingly.

6.2.3.1 Effect of Pruning Thresholds

The first set of curves shows the effect of varying the pruning thresholds ε. In the distributed scheme, the same threshold is used for both the GTC of each node and the information fusion program. The first three criteria shown in Figure 6-1 indicate that, with a low enough

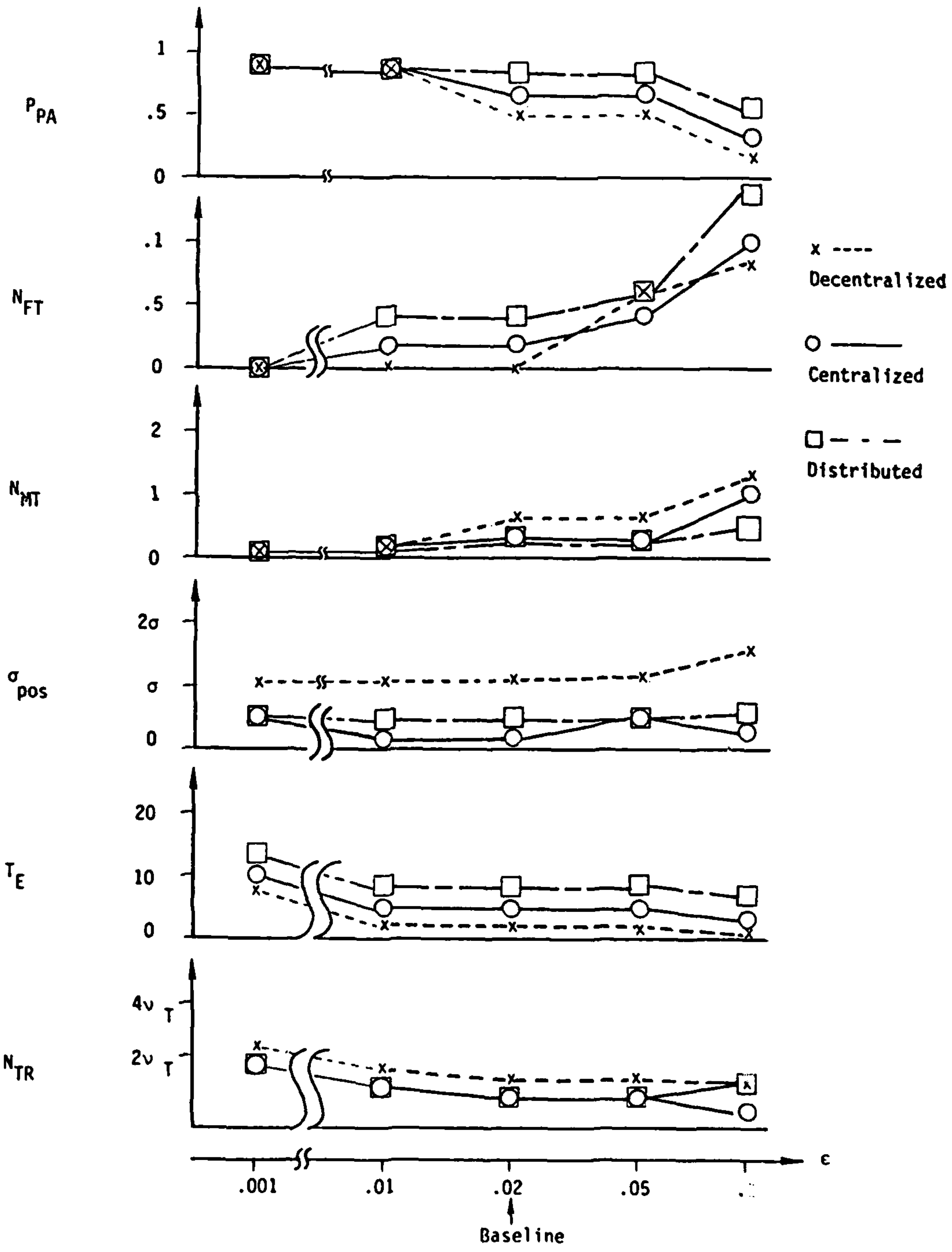

Figure 6-1: Sensitivity to Pruning Threshold

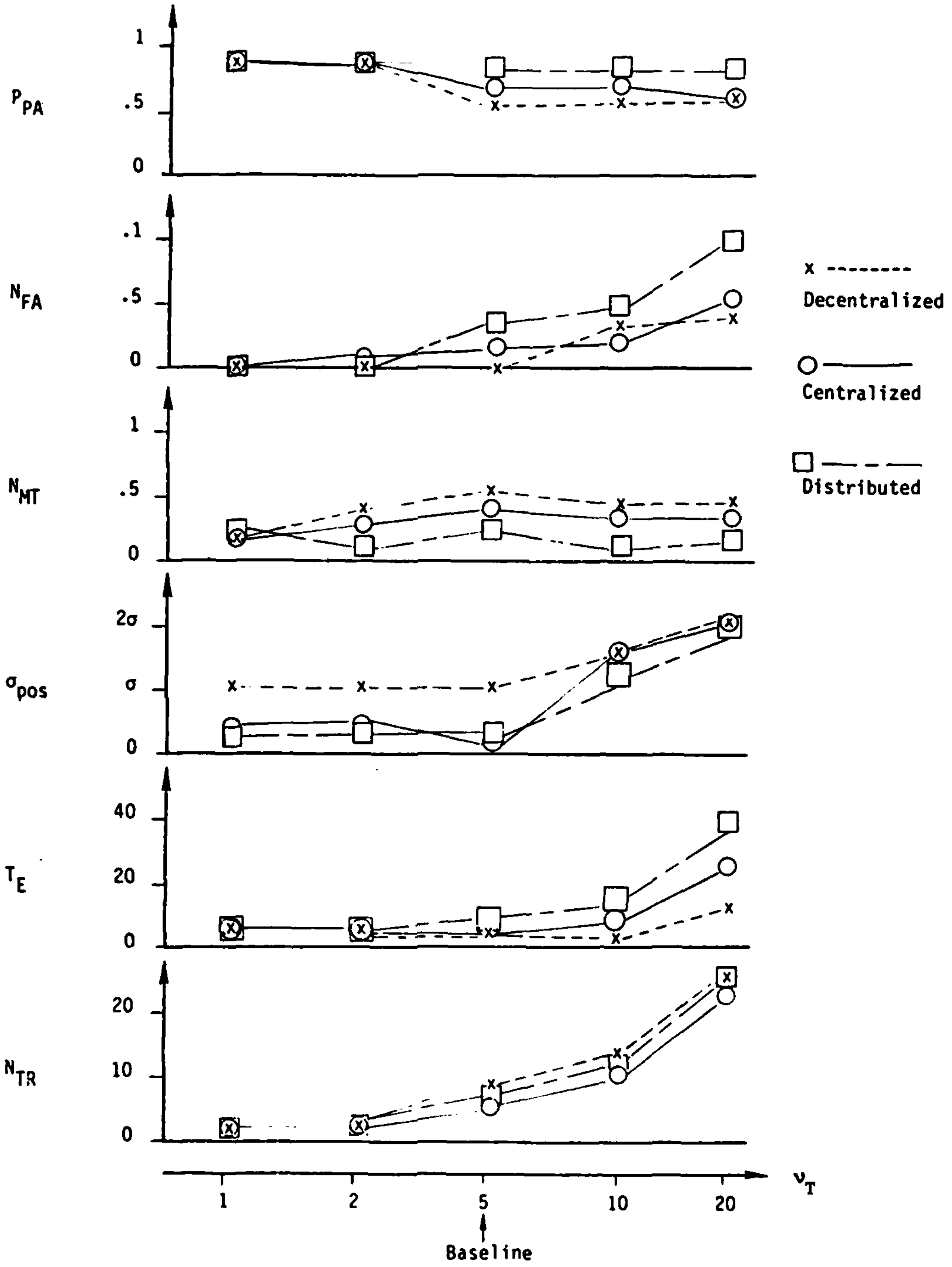

Figure 6-2: Sensitivity to Expected Number of Targets

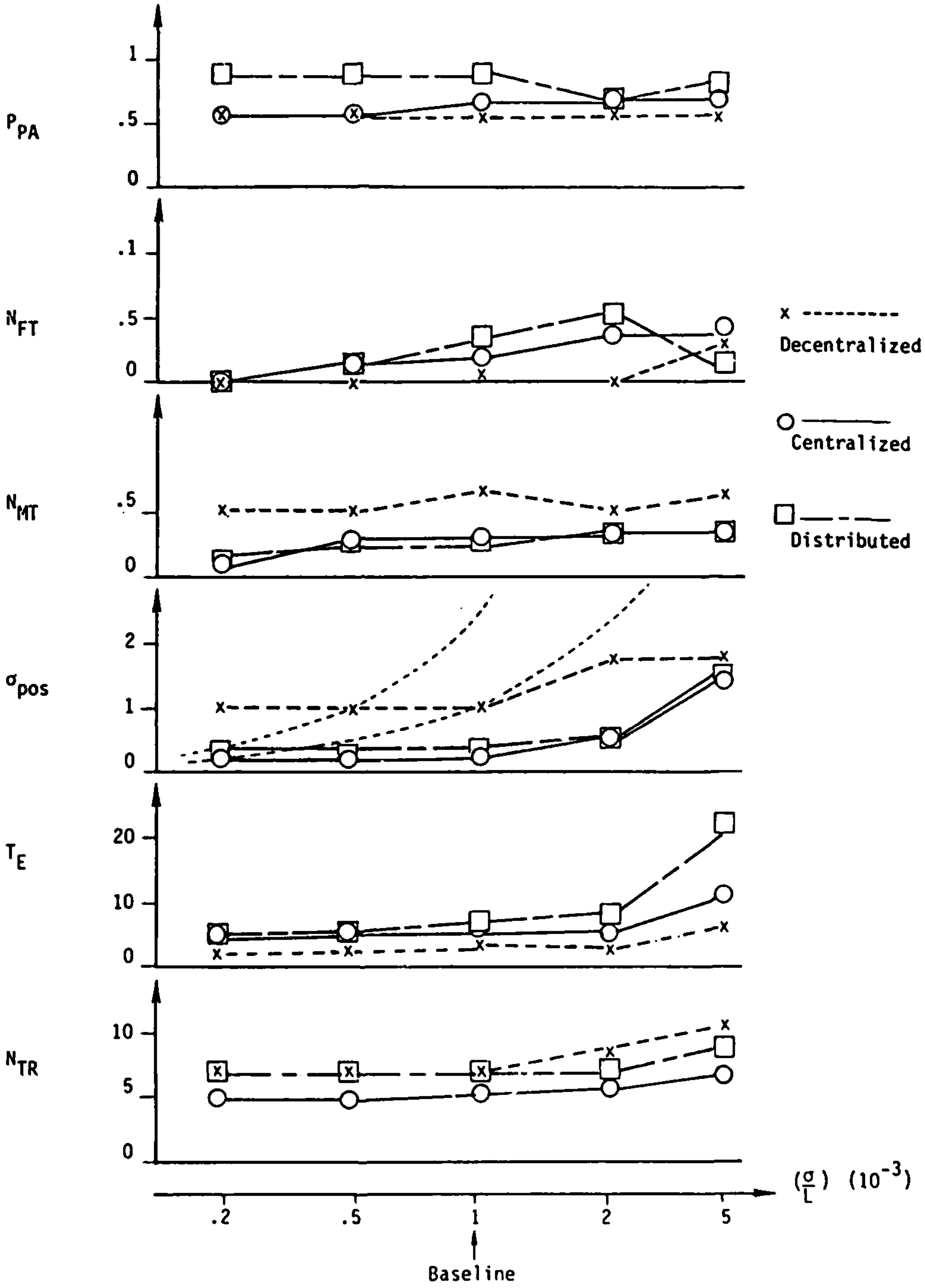

Figure 6-3: Sensitivity to Measurement Error

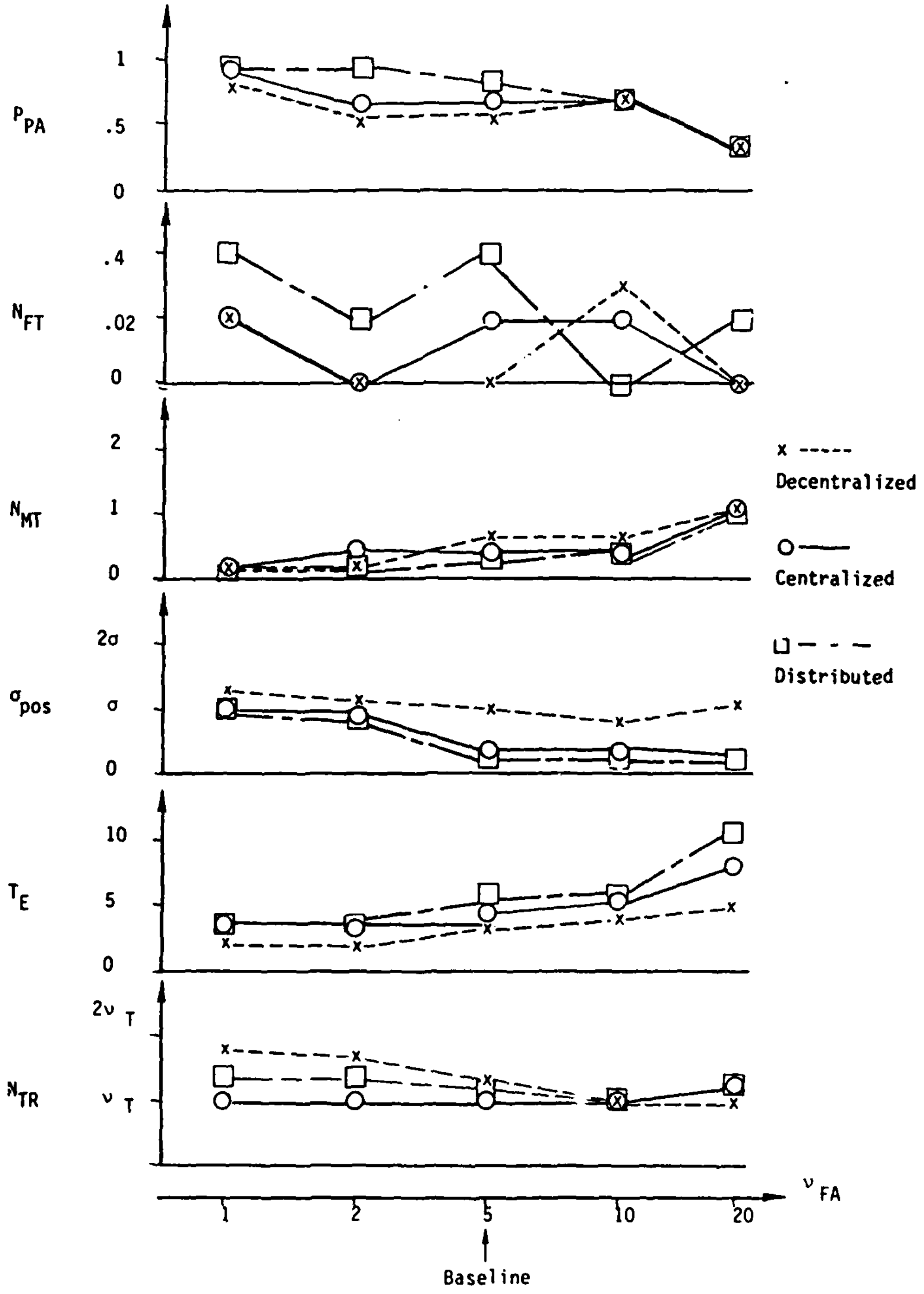

Figure 6-4: Sensitivity to False Alarm Rate

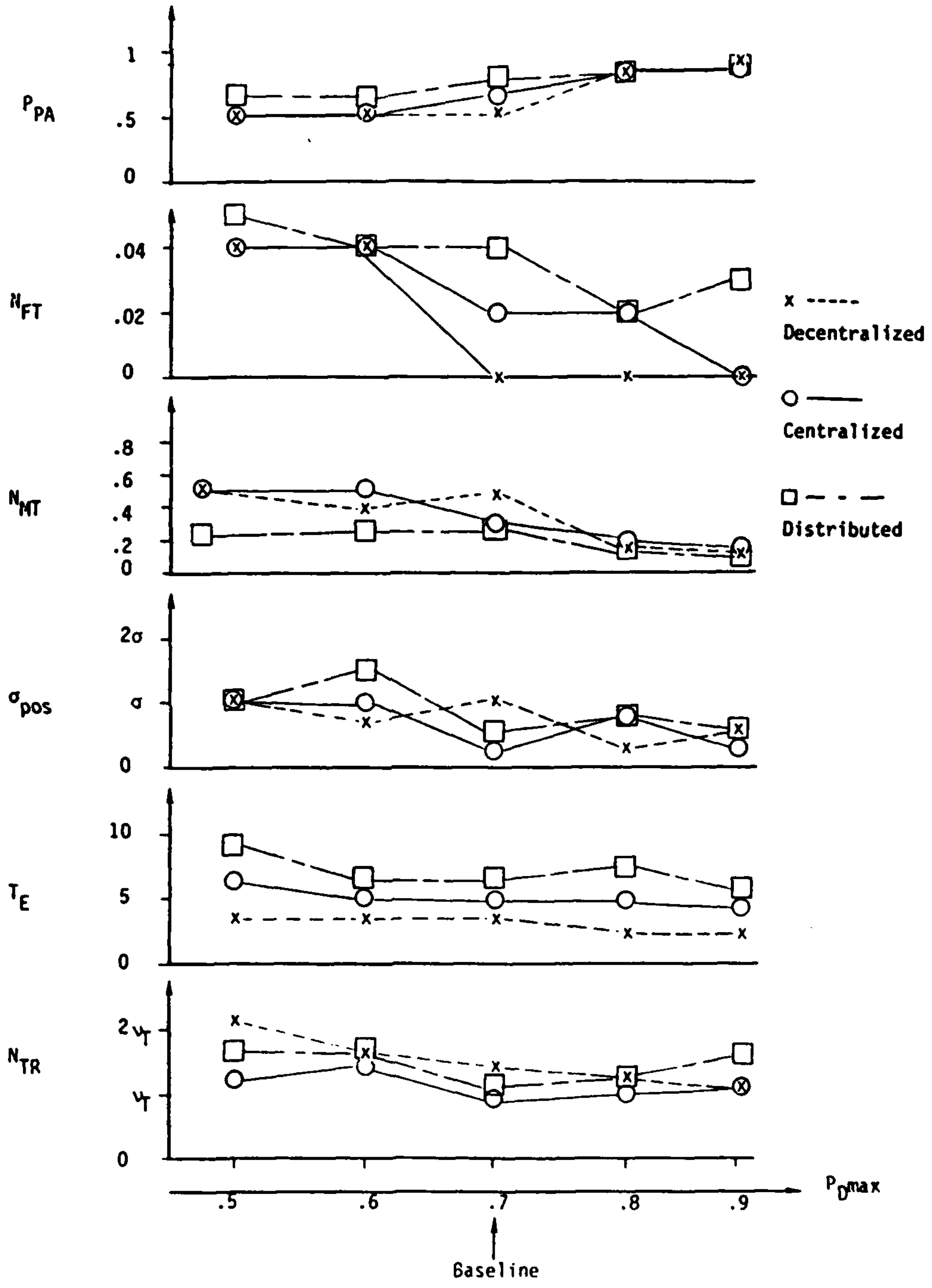

Figure 6-5: Sensitivity to Probability of Detection

pruning threshold, an almost perfect performance is obtained in each of the three schemes. As the last two sets of curves indicate, however, the cost of obtaining this performance varies from scheme to scheme. Almost over the entire range of ϵ, T_E and N_{TR} show the same trend as that in the baseline case result. Namely, the decentralized scheme which utilizes only half the amount of data uses the least amount of CPU time but the largest memory space. The distributed scheme consumes the largest amount of time while its overall average space for tracks is almost the same as the centralized scheme although it maintains two copies of data most of the time.

As the pruning threshold increases, the performance becomes degraded in every scheme. The deterioration is however much more gradual in the distributed scheme than that of the other schemes. From the N_{FT} and the N_{MT} curves in Figure 6-1, it is observed that the performance of all three schemes is biased so that the false-track generation is supressed far more than the missed-target generation. This trend is obvious in the decentralized case and the cases with higher pruning thresholds for every scheme. This bias is caused by the fact that the exact statistics, particularly the expected number of targets and false alarm rates are used to calculate the likelihood functions. By doing so, each scheme tends to falsely dismiss targets by pruning. This aspect will be discussed later. The points of convergence as the threshold approaches zero represent the performance limit of the given sensor systems. Although a slight degradation is observed as the pruning threshold is raised, the relation of the position error statistics of the three schemes remains almost identical.

6.2.3.2 Expected Number of Targets

Figure 6-2 shows the effect of the varying target density. Since the field-of-view is fixed, the expected number of targets is exactly proportional to the density. As the target density increases, the performance generally degrades and the computational requirements increase. Performance degradation is most likely due to the creation of false tracks. These tracks may be created by wrong combination of measurement points. The false track statistics in the high target density region are particularly high in the distributed scheme although its overall performance, P_{PA}, remains superior to the other two schemes. This type of degradation appears probably because the effect of pruning in the fusion program is more crucial than the local GTC program in high target density cases. Since the false alarm rate is kept at the same level, the memory requirement represented by N_{TR} is almost proportional to the target density. The required CPU time increases rapidly, however, in the centralized and the distributed schemes. It is again noteworthy to observe that the distributed case performs better than the centralized scheme. The position estimation deteriorates as the target density increases. This is a result of frequent combining both in the local GTC and the information fusion. This degradation is also caused by the fact that some of the tracks in the best hypotheses are entirely or partially formed by false alarms but, because of high target density, they may be considered "good" enough tracks of actual targets which "happen" to be in their neighborhoods.

6.2.3.3 Measurement Error

The effect of varying the sensor measurement error standard deviation σ is shown in Figure 6-3. In general, as σ increases, the validation regions tend to intersect often so that many clusters become large in size. As a result, there may be a number of equally probable hypotheses in a cluster. As indicated by the N_{TR} curves, the hypothesis combining keeps the number of hypothesis under control. However, in the current implementation of GTC and distributed GTC, all the track-measurement combinations are first exhausted, then hypotheses are pruned, and finally combined after updating every track. Thus, even though the resultant number of tracks is reasonably low the CPU time requirement increases rapidly in accordance with σ. The information fusion program seems particularly susceptible to this kind of explosion of CPU time requirement as indicated by a jump in the T_E curve at $\sigma/L = .005$.

This seems to create a noticeable performance degradation of the distributed scheme for large σ when it seems to lose its definite superiority over the other schemes. The overall detection (tracking) performance, however, remains relatively unchanged. It is hard to judge whether some variation in some criteria. particularly N_{FT}, reflects some significant performance characteristics or they are merely created by random effects. This is so because the sample size is relatively small

(50 runs). On the other hand, the position estimation does not deteriorate considerably as σ increases. In fact, the filtering improvement of estimates is better when σ is large. The effect of the amount of data is again obvious in the σ_{pos} curves although it becomes insignificant when σ is large.

6.2.3.4 False Alarm Rates

The effect of varying the false alarm rates is shown in Figure 6-4. An increase in the false alarm rate has two effects: (1) the increase in the amount of data to be processed, and (2) an increase in the tendency to dismiss tracks. The first effect forces the GTC and the distributed GTC to discard more hypotheses and the second effect results in missed targets, as seen in the P_{PA} and the N_{MT} curves. When the false alarm rates are very high, there may be many occasions where they align so well that they seem to originate from real targets. However, since the false alarms are independent, such events may be still rare. Even if they are not rare, the GTC with non-zero pruning threshold tends to dismiss many measurements as false alarms. Thus, the performance degradation due to the increase in the false alarm rates is the most apparent in the N_{MT} curves. On the other hand, the N_{FT} curves behave rather strangely. This seems to be created by the complicated results caused by the two different factors: aligned false alarms and tendency of dismissing measurements. When the false alarm rates are high, all three schemes have almost identical performance. It is particularly significant to observe that the advantage of having twice the amount of data disappears in such cases. This is interesting because we do not observe such phenomena in conventional filtering or estimation problems. In fact, when the amount of data is doubled, the number of false alarms is also doubled so that the benefit of having additional data diminishes. However, the advantage of the large amount of data is reflected by the σ_{pos} curve. The centralized and distributed schemes are able to select data more accurately without excessive combining as indicated by the σ_{pos} curve. In the decentralized scheme where the amount of data is half, there is a tendency to dismiss measurements frequently, as is apparent from the decreasing N_{TR} curve.

6.2.3.5 Probability of Detection

The last parameter varied is the probability of detection, p_{Dmax}, in Equation (6.2). The result is shown in Figure 6-5. As to be expected, as the probability of detection decreases, the performance deteriorates gradually. The difference in the behavior of the three schemes is however not obvious. In particular, there is no significant difference when the probability of detection is high. The performance is dominated by the missed-target statistics whereas the N_{FT} curves remain at very low level and show only random changes. The position estimate errors behave rather strangely. This again reflect the fact that, when the origin of each measurement is not certain, the amount of data is sometimes advantageous but not always. As seen in the N_{TR} curves, when the number of scans is small (the decentralized scheme), the low probability of detection requires more hypotheses and thus more memory space. However, for the centralized and the distributed schemes where the number of scans is twice as many as the decentralized scheme, this trend is not so obvious. The CPU time requirement increases as the probability of detection decreases in every scheme. However, the relation among the different schemes remains substantially unchanged.

6.2.4 Summary of Results and Supplemental Discussions

The observations from the comparative statistics obtained by the varying key parameters may be summarized as follows:

1. In general, as the parameters move to more difficult values, the performance degrades accordingly in every scheme. In most cases, however, the relative performance displayed in the baseline case of the three schemes remains unchanged.

2. When the external conditions are severe (e.g., with high target density, high false alarm rates, and large measurement errors), the difference in performance sometimes becomes less obvious. However, there is no cross over in the performance ordering of the three schemes.

3. The advantage of using less data (the decentralized scheme) appears only in saving the CPU time. Under mild external conditions, the decentralized scheme requires more memory space than the other schemes.

4. The CPU time requirement increases rapidly with high target density, high false alarm rates, large measurement error and low pruning threshold, both in the centralized and the decentralized schemes.

In order to evaluate the performance, we have used the two demerit indices, N_{FT} and N_{MT}. In the baseline comparison and the subsequent sensitivity studies, we have occasionally observed some bias toward more N_{MT}, i.e., a trend to dismiss targets than creating false tracks. These two indices correspond exactly to the two basic sensor parameters, the false alarm rate and the probability of false dismissal (1 - prob. of detection). Thus, like the false alarm rate and the probability of false dismissal, N_{FT} and N_{MT} are complementary. For example, when any pruning scheme is employed, there is always some chance of dismissing a "true" hypothesis consisting of true tracks of the real targets. This possibility has not been considered in deriving our basic hypothesis evaluation equations. Although how to take care of this is an open theoretical problem, the pair (N_{FT}, N_{MT}) may be affected by changing some parameters used in the likelihood functions, such as the a priori target density, the probability of detection and the false alarm rates. The last analysis in this section shows how the two indices change when the a priori target density is modified from its true value.

Figure 6-6 shows the performance variation when the expected number of targets used in the algorithm is modified to ν_T^M from its true value ν_T. As expected, N_{FT} increases and N_{MT} decreases as ν_T^M increases. An index representing the overall performance, P_{PA}, has a peak in every scheme. The peak is however at a different point for each scheme. The determining factor seems to be the amount of data. Namely, when the amount of data is large, the chance of missing targets is low so that there is less need for large compensation. The increase of ν_T^M, however, forces the GTC and the distributed GTC to maintain more hypotheses and to use more computation time.

Figure 6-7 shows the two dimensional change of (N_{FT}, N_{MT}) according to ν_T^M. The curves in this figure may be regarded as trade-off curves.

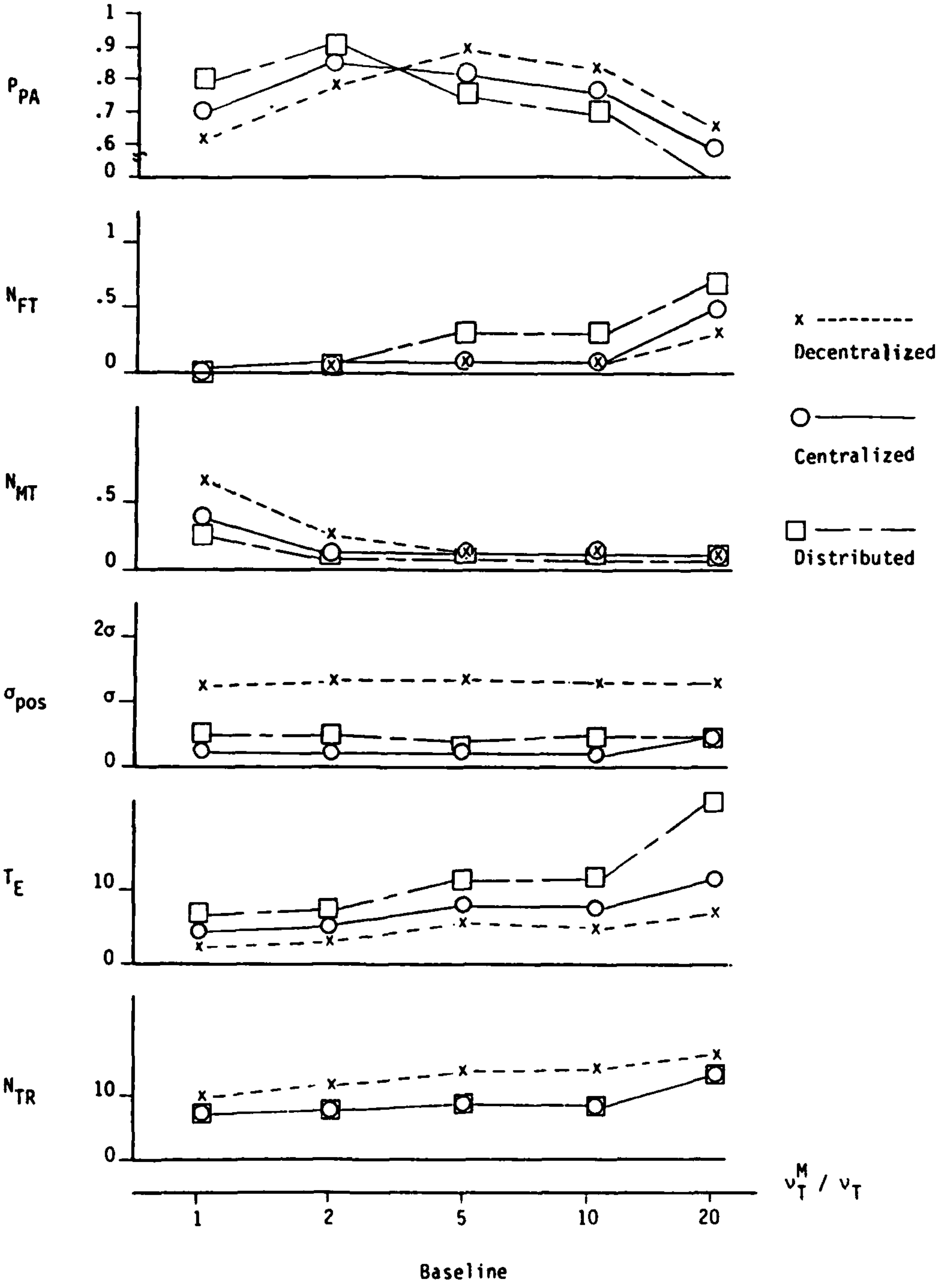

Figure 6-6: Sensitivity to Error in Expected Number of Targets

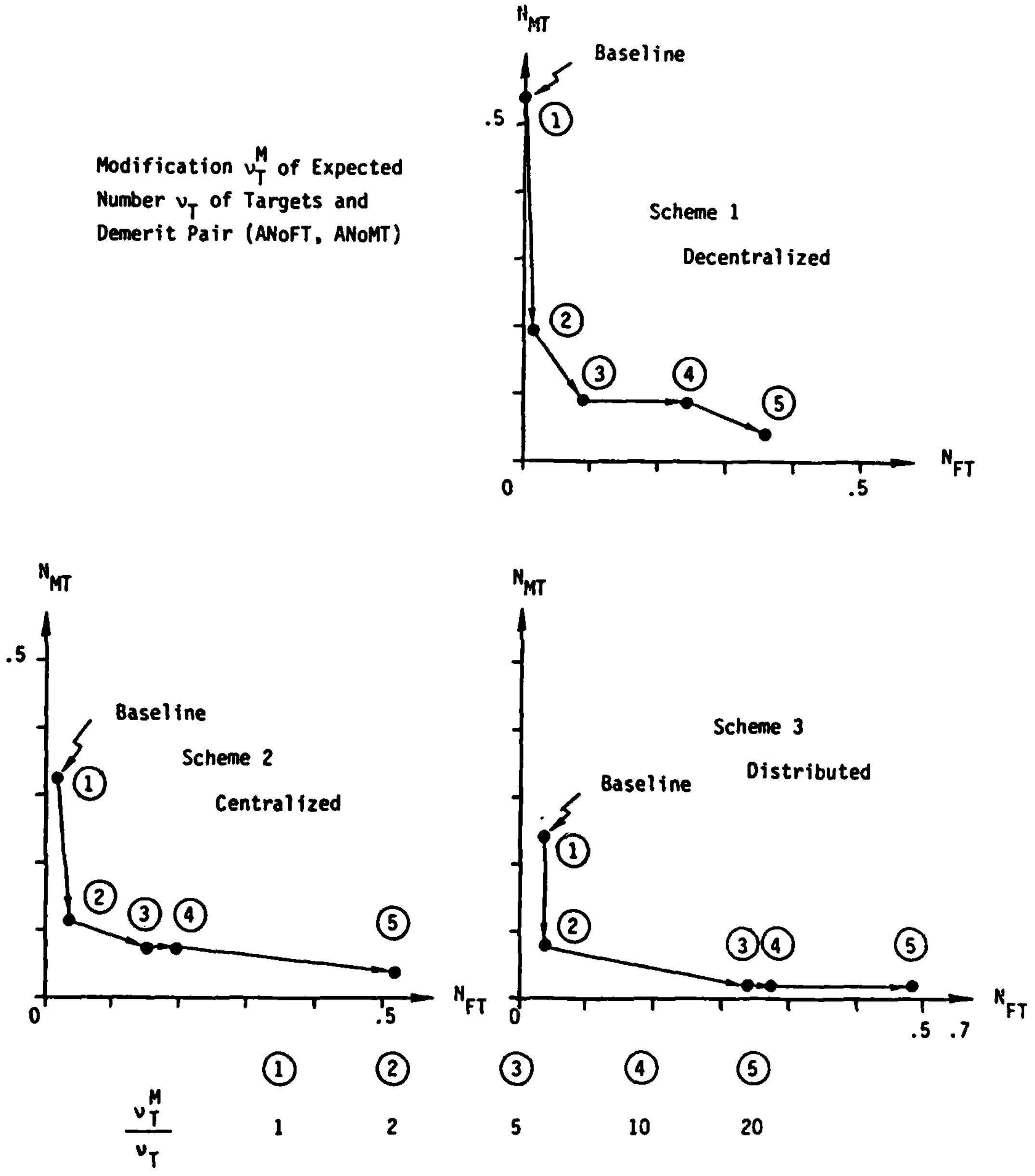

Figure 6-7: Two-Dimensional Change of False and Missed Tracks

They may help the user of the GTC or the distributed GTC in choosing the right parameters. Other parameters which may be modified are the false alarm rates and the probability of detection. Moreover, the calculation of the likelihood functions for newly detected targets can also be modified. Which parameters should or should not be modified is still an open question.

6.3 AN ALMOST CONSTANT VELOCITY TARGET EXAMPLE

In this section we consider a class of dynamic target models where the velocity is almost constant. Such models can approximate targets which do not maneuver during the sensor observation interval.

6.3.1 Example Scenario

Almost constant velocity target models are next in complexity to the simplest target model, i.e., the stationary target model described in Section 6.2, but are commonly used in the multitarget tracking literature. By an almost constant velocity target model, we mean a dynamical model in which the position and the velocity constitute a state and the time-derivative of the velocity is a white noise vector with small intensity. In this section, we shall explore a one-dimensional almost-constant-velocity target model, in which each target state x(t) at time t has a dynamical model given by:

$$x(t) = \begin{bmatrix} u(t) \\ \\ v(t) \end{bmatrix} = \begin{bmatrix} u_0 + \int_{to}^{t} v(\tau)d\tau \\ \\ v_0 + \ w(t) \end{bmatrix} \tag{6.5}$$

where u(t) is the position, v(t) the velocity at time t, and w(.) is a Wiener process with Var.(w(Δt)) = QΔt. We assume two nodes each of which has a sensor measuring the velocities of targets as well as their positions. The node layout is shown in Figure 6-8. The initial velocity of each target is assumed to be much larger than the intensity of the white noise which excites the state, and thus targets always move in the same directions. Therefore, we assume that each sensor has a preprocessor which filters through only moving objects and divides all the measurements by their directions into two separate sets and that we do not need to associate measurements across the two sets. Moreover, when each target is viewed as an object moving in a two-dimensional space (position, velocity), targets never cross each other although they may pass each other. For this reason, in our experiments we assume that targets move only in one direction.

As shown in the figure, the two sensors have non-overlapping fields-of-view. The distance between the two fields-of-view is assumed to be the same as the length of the field-of-view of each sensor. The two sensors have identical characteristics. The detection probability for each sensor n is given by:

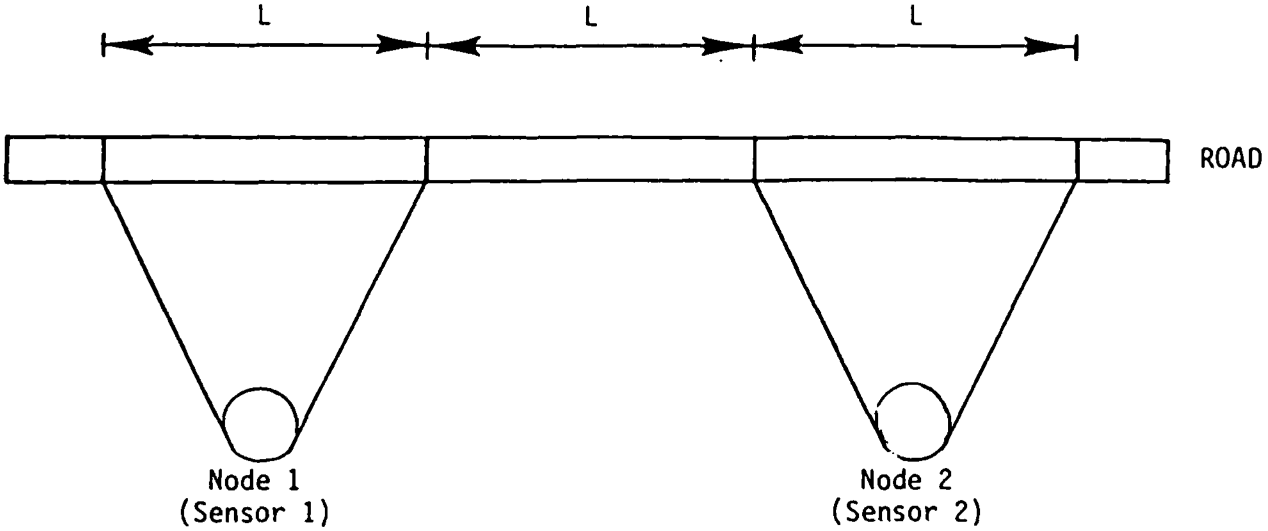

Figure 6-8: Node Layout

$$
\begin{aligned}
p_D^{(n)}(x) &= p_D^{(n)}(u,v) \\
&= p_{D\max} \int_{U^{(n)}} g(u_m-u;R_u)du_m \int_{V} g(v_m-v;R_v)dv_m
\end{aligned}
\qquad (6.6)
$$

where

$$g(\xi;R) = (\sqrt{2\pi R})^{-1} \exp(-\tfrac{1}{2}\xi^2/R) \qquad (6.7)$$

is the one-dimensional Gaussian density function and

$$
\begin{aligned}
U^{(n)} &= [u_{\min}^{(n)}, u_{\max}^{(n)}] = \text{field-of-view of sensor } n \\
V &= [v_{\min}, v_{\max}] = \text{velocity interval observable (same for both sensors)}
\end{aligned}
\qquad (6.8)
$$

The measurement error probability for sensor n is given by:

$$p_M^{(n)}(y|x) = p_M^{(n)}(u_m, v_m | u,v)$$

$$= \frac{g(u_m - u; R_u)\ g(v_m - v; R_v)}{\int_{U^{(n)}} g(\xi - u; R_u) d\xi \int_V g(\eta - v; R_v) d\eta} \tag{6.9}$$

By choosing the left end of the field-of-view of sensor 1 as the origin, we have:

$$u_{min}^{(1)} = 0, \qquad u_{max}^{(1)} = L,$$
$$u_{min}^{(2)} = 2L, \text{ and } u_{max}^{(2)} = 3L. \tag{6.10}$$

Thus, the sensor performance is determined by p_{Dmax} in (6.6) and the measurement accuracies

$$\sigma_u = \sqrt{R_u} \quad \text{and} \tag{6.11}$$

$$\sigma_v = \sqrt{R_v} \ . \tag{6.12}$$

The initial velocities of the targets have the distribution shown in Figure 6-9. The initial positions were assumed to be uniformly distributed on an interval U_o which is chosen so that we could expect a

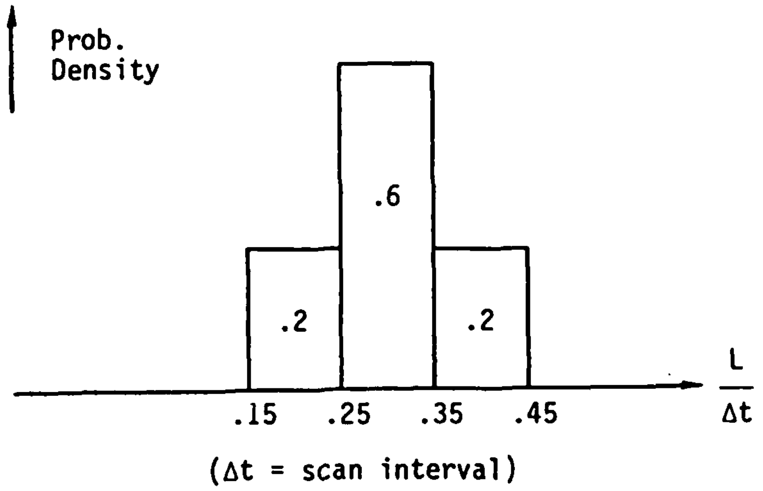

Figure 6-9: A Priori Distribution of Initial Velocity

target at any position on [0, 3L] when the first scan was made and a new target could appear at the last scan. We further assumed that the two sensors were synchronized with the identical scan interval Δt and that $v_{min} = (.15)\ (L/\Delta t)$ and $v_{max} = .45\ (L/\Delta t)$ so that (6.6) - (6.9) made sense. The number of false alarms from each sensor was assumed to be Poisson with mean ν_{FA}, and each false alarm is distributed uniformly on $U^{(n)} \times V$.

The baseline conditions chosen for this example are shown in Table 6-5.

Table 6-5: Baseline Parameters for Almost Constant Velocity Case

Pruning Threshold	ε	.02
Target Density	β_T	0.5/L
Position Measurement Error	σ_u	$10^{-3}L$
Velocity Measurement Error	σ_v	$10^{-3}L/\Delta t$
Expected Number of FA	ν_{FA}	4/scan
Probability of Detection	P_{Dmax}	.7

6.3.2 Communication Schemes and Performance Measures

The three communication schemes introduced in the previous section (decentralized, centralized and distributed) were examined. The simulation time is chosen to be 10 scan intervals. In the decentralized scheme, each node processes only the data available to it. In the centralized scheme, all the data is processed in a centralized manner either by a single processing node or redundantly by two nodes with every data set exchanged at each scan between the two nodes. On the other hand, in the distributed scheme, the communication between nodes is much less frequent and is at the hypothesis level rather than the sensor measurement level. In this example, the distributed scheme communicates every 5 scans, i.e., only twice in the total length of scenario.

Since in this example the targets move only in one direction (left to right), we cannot compare the node performance in the decentralized scheme with other schemes using the same performance measures of the previous example. For example, the downstream node 2 may have a chance of seeing only half of the targets getting into the region and the role of the upstream node 1 is asymmetric with respect to the other node 2. For this reason, we evaluate the decentralized scheme by assuming an additional ad hoc coordination system which integrates the outputs of the two nodes. First, the best hypothesis is extracted from each node. Tracks in the best hypotheses from the two nodes are then tested by a thresholding modified rectangular Munkres algorithm [13]. A track τ_1 from node 1 and a track τ_2 from node 2 are declared to originate from a single real target when the distance between the best estimates of the position-velocity pairs is less than a given threshold. Let τ_1 and τ_2 be the hypotheses from the two nodes in the decentralized scheme and T_{ST} be the set of pairs (τ_1, τ_2) which are judged to originate from a single target as described above. Then, we consider the following two sets of tracks as outputs of the two different coordination systems:

(1) AND-LOGIC SIMPLE FUSION

$$\{\tau_1 \cup \tau_2 | (\tau_1, \tau_2) \in T_{ST}\}$$

(2) OR-LOGIC SIMPLE FUSION

$$\{\tau_1 \cup \tau_2 | (\tau_1, \tau_2) \in T_{ST}\}$$

$$\cup \{\tau_1 \in \lambda_1 | \text{there is no } \tau_2 \text{ such that } (\tau_1, \tau_2) \in T_{ST}\}$$

$$\cup \{\tau_2 \in \lambda_2 \text{ there is no } \tau_1 \text{ such that } (\tau_1, \tau_2) \in T_{ST}\}$$

The reason that we have named such coordination as "simple" is apparent. as well as the adjectives "and-logic" and "or-logic". The introduction of these coordination systems for the decentralized scheme serves a two-fold objective: to test the effect of the frequency of communication at the hypothesis level; and to see how well or poorly such simple "heuristic" operations as described above perform when compared with more well-defined outputs from the centralized and the distributed schemes.

The same set of performance criteria as those described in Section 6.2.2 is used; N_{FT}, N_{MT}, P_{PA}, σ_{pos}, T_E and N_{TR} shown on Table 6-2 are used to evaluate the performance of every scheme. Since we have the velocity component in a target state and the velocity measurement in every sensor measurement, an additional criterion, i.e., the velocity estimation error σ_{vel} is included. The evaluation procedure for each scheme is the same as that described in Section 6.2.2.

6.3.3 A Sample Run

In this section we describe the results of a single sample run. First we describe the data from this run, and next we summarize the operation of each of the three schemes.

6.3.3.1 Sample Run: Data

Figure 6-10 shows a typical sample of the 10-scan position measurements from two nodes under the baseline condition. There are 73 measurements in total from the two nodes (sensors) in the 10 scans. Out of these 73 measurements, only ten originate from the real targets while the rest are false alarms. As seen in this figure. the nominal condition can be characterized as a high-false-alarm-rate, low probability-of-detection case which makes the tracking difficult. From these data. it is almost impossible to extract tracks by human eye. Of course, this is true particularly because Figure 6-10 shows only position measurements. Figure 6-11 shows the same sample where velocity measurements are attached. In this figure, each velocity measurement is represented by a unit-length arrow originating from its position measurement and pointing to the expected position at the next scan time. Thus an arrow pointing straight down indicates zero velocity and an arrow pointing to the right indicates infinite velocity. With the high false alarm rate. the relatively low probability of detection and the existence of the masked region, it is still very difficult for human eye to extract target trajectories.

Figure 6-12 shows the true target trajectories of this sample, and Figure 6-13 is the superposition of the measurement data (Figure 6-11) and the true target trajectories (Figure 6-12). Target 1 appears in the masked region and moves eventually into the field-of-view of sensor 2 where it has the opportunity of being detected four times, but is actually detected only twice. The rest of the targets, 2, 3, and 4, appear

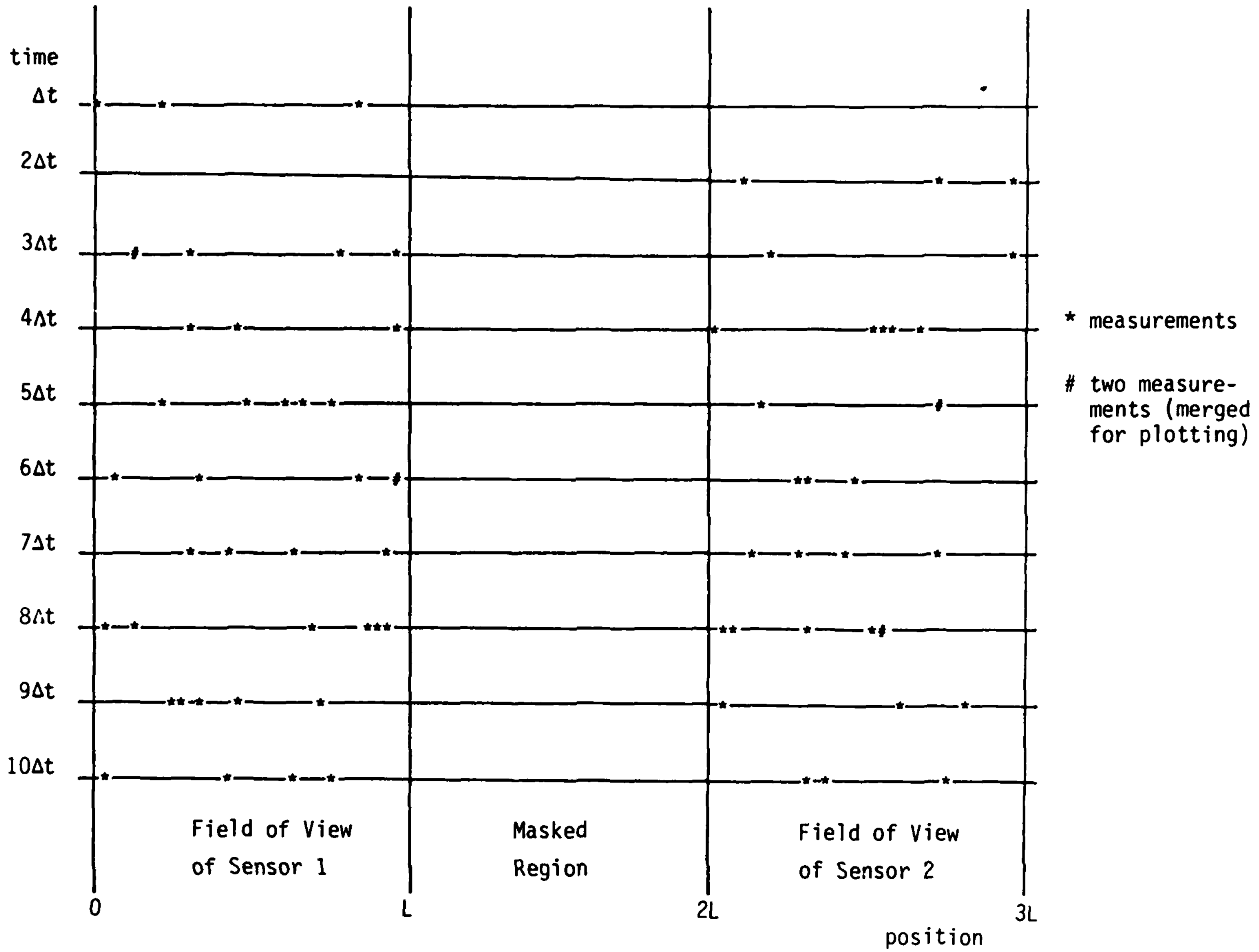

Figure 6-10: A Sample of 10 Scans of Position Measurements

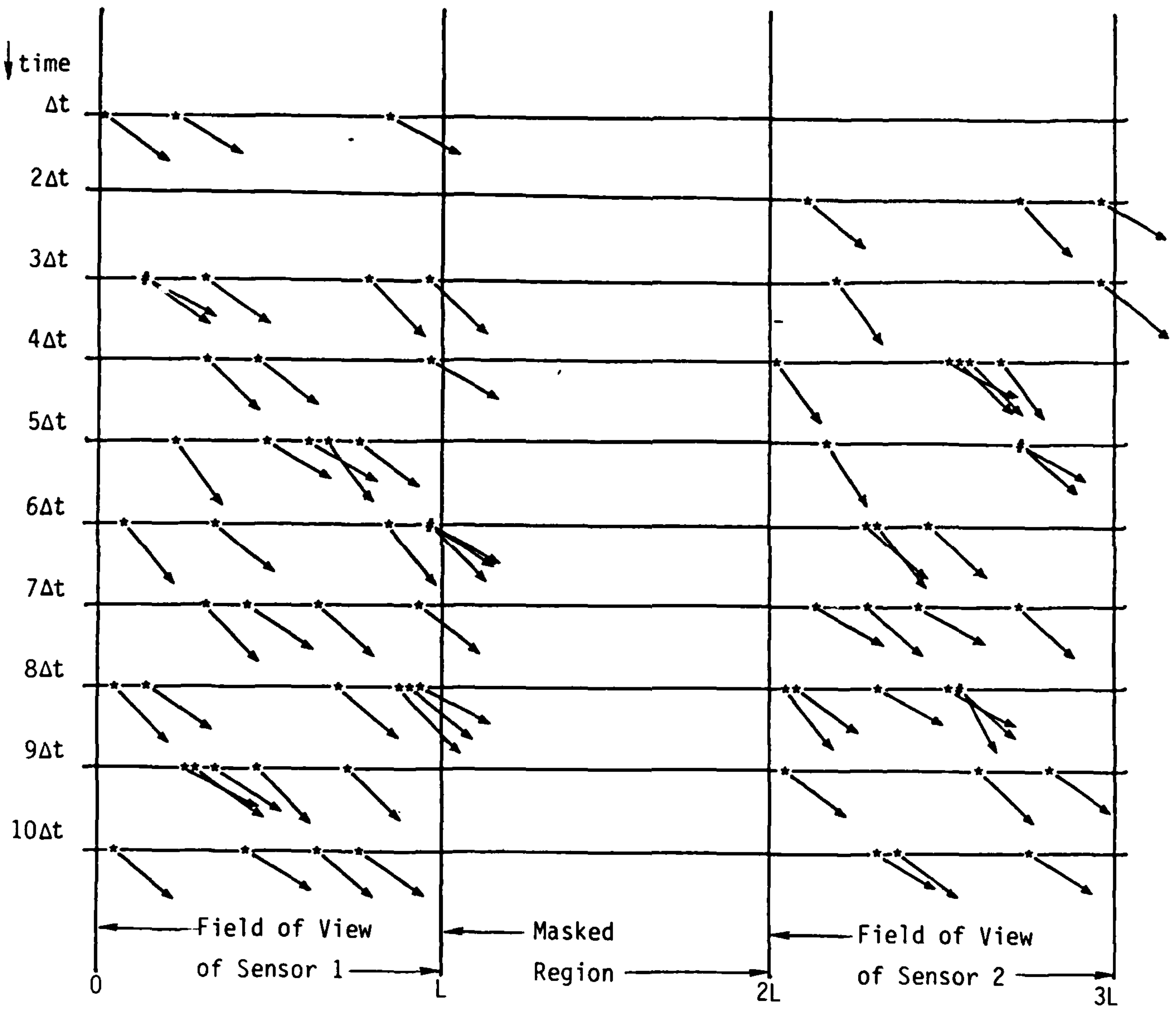

Figure 6-11: A Sample of 10 Scans of Position and Velocity Measurements

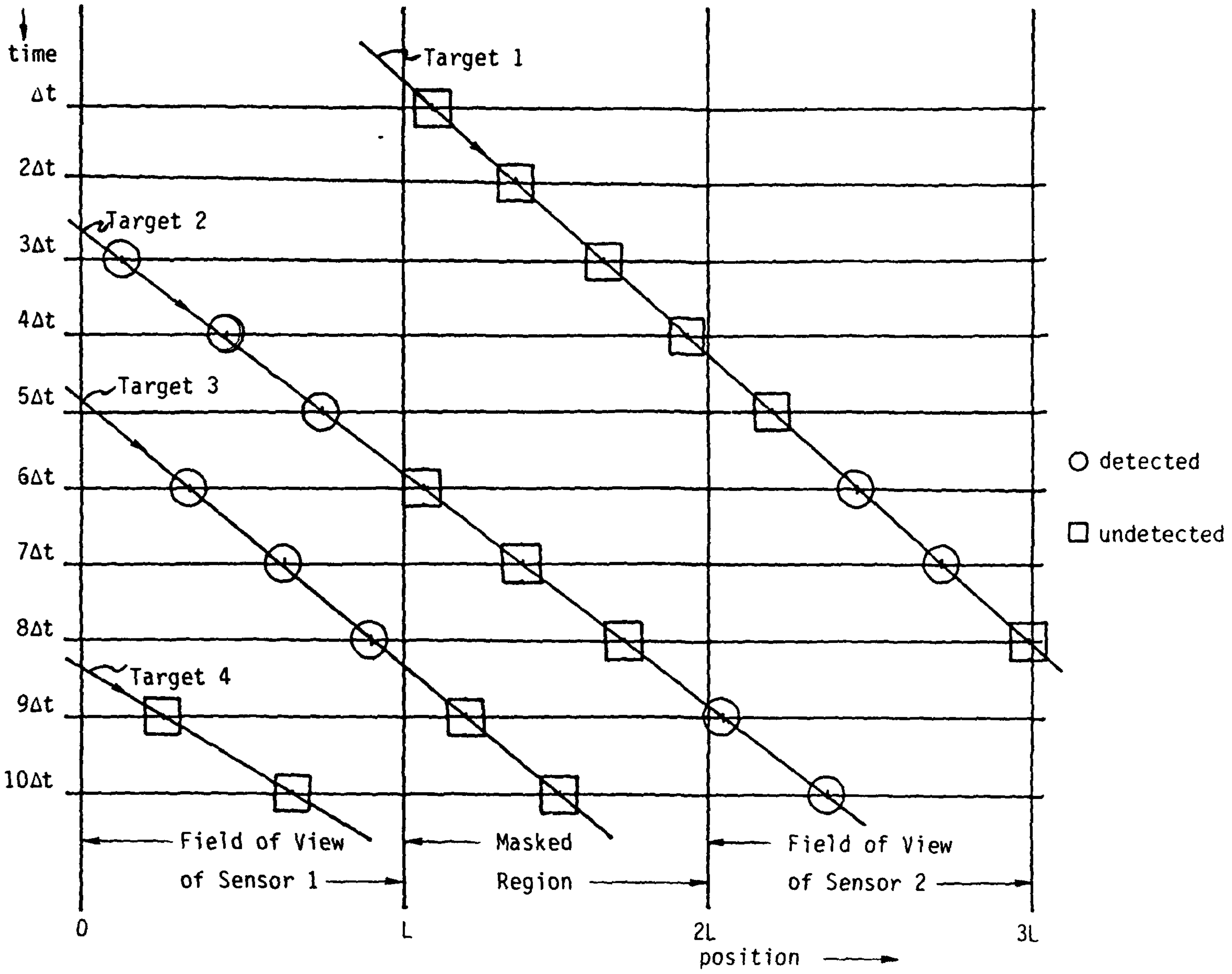

Figure 6-12: True Target Trajectories of the Sample

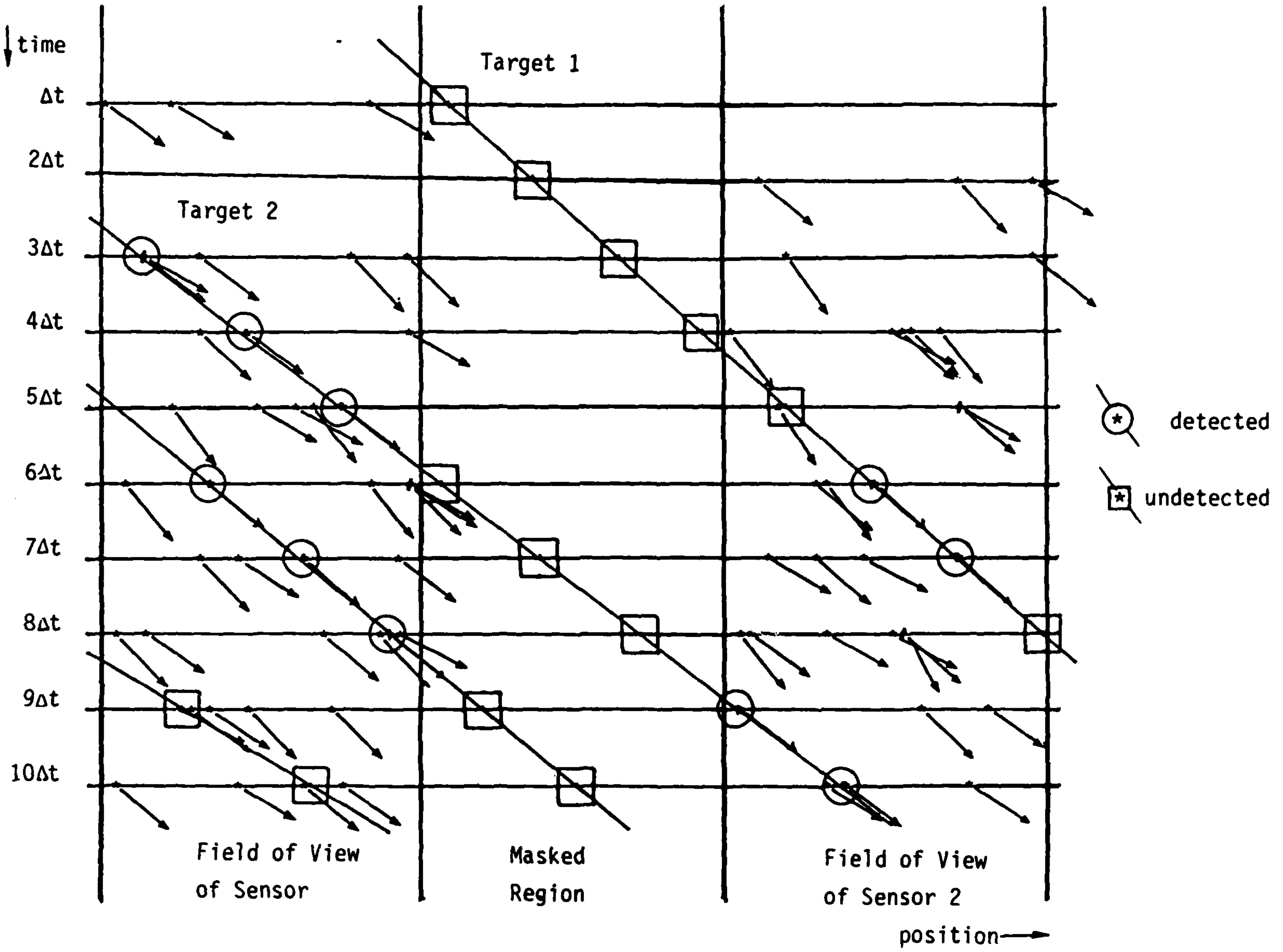

Figure 6-13: Measurement Data and True Target Trajectories

in the field of view of Sensor 1. Only target 2 moves across into the field-of-view of Sensor 2. Targets 2 and 3 are detected whenever they have a chance of being detected. On the other hand, target 4 is never detected.

6.3.3.2 Sample Run: Decentralized Scheme

At the end of Scan 5, node 1 confirms target 2, i.e., all the hypotheses contain the track of the three measurements while the other targets have not yet been detected. By Scan 10, target 3 has been confirmed by node 1, and targets 1 and 2 have been confirmed by node 2. Thus, at the end of the sample run, the best hypothesis contains three tracks corresponding to targets 2 and 3 in node 1, and targets 1 and 2 in node 2. Consequently, the simple AND-logic fusion should form one track corresponding to target 2 while the simple OR-logic fusion should form three tracks corresponding to targets 1 to 3 if the fusion mechanisms work properly. However, due to the particular value for the threshold used by the Munkres assignment algorithm, the two tracks which should be identified are not judged to originate from the same track. As a result, the simple AND-logic fusion forms no track, i.e., no false track but four missed targets, while the simple OR-logic fusion forms four tracks, i.e., one false track and one missed target.

6.3.3.3 Sample Run: Centralized Scheme

At the end of the last scan, there are 18 clusters. Three of them have confirmed tracks which correctly correspond to the real targets, 1 to 3. All other clusters have only tentative tracks with small probabilities. Thus, there is no false track and one missed target. Since the missed target is undetected by all the sensors, there is no chance for it to appear in the output of any scheme.

6.3.3.4 Sample Run: Distributed Scheme

Up to scan 5, everything works in exactly the same way as in the decentralized scheme, i.e., target 2 has been confirmed by node 1 while other targets have not been detected yet and all other tracks are tentative. At the end of scan 5, the information, in terms of clusters, hypotheses and tracks, from the two nodes is "fused". In the first step of this fusion, the track-to-track likelihood function is calculated for each pair (τ_1, τ_2) of tracks from nodes 1 and 2. It turns out that the only feasible (positive likelihood) pairs are in the form of either (τ_1, ϕ) or (ϕ, τ_2). This reflects the true situation that no target has appeared in both of the fields-of-view. One of the tentative tracks from node 1 indicates that, if it had originated from a real target, it should have been detected also by node 2. Since it has not been detected, this tentative track is rejected. Thus, out of 7 possible tracks from node 1 and 8 tracks from node 2, one (correctly) confirmed track and 13 tentative tracks become common tracks to both nodes.

Node 2 correctly extends the confirmed track when the target enters the field-of-view of sensor 2. With respect to the other targets, 1, 3 and 4, the distributed scheme performs in a way very similar to other schemes until the last scan after which the second information fusion is performed. At the end of the last scan (before the second fusion), several of previously fused tracks have been eliminated by each node. There are ten new tracks (including the confirmation of target 3) from node 1 and 2 tracks (including the confirmation of target 1) from node 2. When the track-to-track likelihood function is calculated, four previously fused tracks are eliminated because one of the nodes has rejected them. Subsequently, the confirmed and previously fused track is fused again with new information being provided by node 2, while the other five previously fused tracks have not gained any additional information because any real target is not expected to have appeared within the field-of-view of any sensor. Thus, the information(clusters, hypotheses and tracks) obtained by this second fusion is the same as that obtained by the centralized scheme except for small errors in track distribution parameters, track likelihood and probabilities introduced by approximations in the information fusion program. In this sample run, the performance of the centralized scheme is almost identical to the distributed schemes. This means that the hypothesis management used in the centralized GTC, the distributed GTC's and the information fusion program have not altered the objective of the distributed system, i.e., to produce the performance of a centralized system while distributing the tracking tasks among the two nodes.

6.3.4 Monte Carlo Simulation Results

Although what is discussed above is only a sample, it turns out that this performance comparison of the centralized and the distributed schemes is quite typical at least for the baseline case. This result is in contrast to that of the previous section where we saw a slight advantage of the distributed scheme over the centralized scheme. The similar performance of the two schemes is probably due to two factors. First, only a small fraction of targets appear in the field-of-view of the sensor because of the relatively large masking. Second, false alarms do not disturb the performance greatly despite the relatively high density because the velocities in the measurements serve as strong discriminants. Table 6-6 shows the base case comparison of the four schemes obtained by 100-run Monte Carlo simulations.

It is obvious from Table 6-6 that the two decentralized schemes perform worse than the centralized and the distributed cases. This result shows a clear advantage of frequent communication over less frequent communication and of rigorous fusion algorithms over heuristic fusion algorithms. As could be expected, the simple AND-logic scheme's performance is the worst because of its extreme cautiousness. Since there is little chance that both nodes confirm the same target, this scheme tends to create many missed targets. On the other hand, the simple OR-logic performs almost as well as the centralized and the distributed schemes with respect to the missed target statistics. However, since this scheme accepts non-agreed-upon tracks rather blindly, the number of false tracks is exceptionally high. However, when we view

Table 6-6 Baseline Comparison for Dynamic Targets

	P_{PA}	N_{FT}	N_{MT}	σ_{pos}^{*}	σ_{vel}^{**}	T_{E}^{***}	N_{TR}
	Prob. of Perfect Association	Average No. of False Tracks	Average No. of Missed Targets	Posiiton Estimation Error (*)	Velocity Estimatio Error (**)	Average Execution Time (***)	Average No. of Tracks
Decentralized (Simple AND Fusion)	.09	0	2.97	8.91	3.53	6.42	26.3
Decentralized (Simple OR Fusion)	.29	.27	1.08	6.06	1.91	6.42	26.3
Centralized	.34	.05	1.04	5.29	1.70	10.93	20.1
Distributed	.34	.06	1.04	5.29	1.70	11.82	32.2

(*) Normalized by the standard deviation of the position measurement error.

(**) Normalized by the standard deviation of the velocity measurement error.

(***) Average execution time for one run:

Decentralized:	20 Data Sets
Centralized:	20 Data Sets
Distributed:	20 Data Sets + 2 Fusion Operations

P_{PA}, the probability of perfect association, as a single proxy of performance for each scheme, the simple OR-fusion decentralized scheme performance is quite close to the centralized and distributed schemes.

The comparison of performance between the centralized and the distributed schemes is not conclusive since both perform equally well. The trend in the computational requirement is very similar to that we have observed in the previous example. The decentralized system uses much less CPU time but requires more space. The distributed scheme requires more time and space than the centralized scheme. The average estimation error is comparable to the measurement error for the velocity and is about 5 to 9 times bigger for the position. This is so because the average number of measurements in a track is very small due to the low probability of detection and the large masking. The position estimation errors are further worsened since the random change in the velocity is accumulated and added to the positional uncertainty.

Comparative statistics were obtained by 100-run Monte Carlo runs For each of the six key parameters (i.e., the pruning threshold, the target density, position measurement error, velocity measurement error, false alarm rate and probability of detection). The parameter values used in these tests are summarized in Table 6-7. The results are summarized in Tables 6-8 to 6-13.

Table 6-7 Parameter Values for Sensitivity Study

Parameter	Values		
	Low	Baseline	High
Pruning Threshold - ε	0.001	0.02	0.1
Target Density - ν_Γ	0.25/L	0.5/L	5/L
Position Meas. Error - σ_u	0.0001L	0.001L	0.01L
Velocity Meas. Error - σ_v	0.0001L/Δt	0.001L/Δt	0.01L/Δt
False Alarm Rate - ν_{FA}	2	4	8
Prob. of Detection - P_{Dmax}	0.5	0.7	0.9

Table 6-8 Sensitivity to Pruning Threshold

ε	P_{PA}	N_{FT}	N_{MT}	σ^{*}_{pos}	σ^{**}_{vel}	T^{***}_{E}	N_{TR}
- Decentralized (Simple AND Fusion) -							
.001	.08	0	3.01	10.21	3.94	12.00	44.2
.02	.09	0	2.97	8.91	3.53	6.42	26.3
.1	.07	0	3.02	12.03	3.10	.87	2.8
- Decentralized (Simple OR Fusion) -							
.001	.21	.56	1.22	6.12	1.86	12.00	44.2
.02	.29	.27	1.08	6.06	1.91	6.42	26.3
.1	.12	.30	2.13	7.49	2.22	.87	2.9
- Centralized -							
.001	.38	.11	.89	5.43	1.80	30.55	40.4
.02	.34	.05	1.04	5.29	1.70	10.93	20.1
.1	.11	.01	2.23	5.85	2.01	1.17	2.2
- Distributed -							
.001	.39	.08	.90	5.42	1.80	33.43	62.8
.02	.34	.06	1.04	5.29	1.70	11.82	32.2
.1	.11	.01	2.23	5.85	2.01	1.17	3.2

(*) Normalized by the standard deviation of the position measurement error.

(**) Normalized by the standard deviation of the velocity measurement error.

(***) Average execution time for one run.

Table 6-9 Sensitivity to Target Density

ν_T	P_{PA}	N_{FT}	N_{MT}	σ_{pos}^{*}	σ_{vel}^{**}	T_E^{***}	N_{TR}
- Decentralized (Simple AND Fusion) -							
.25/L	.32	0	1.37	8.84	2.06	2.08	9.4
.5/L	.09	0	2.97	8.91	3.53	6.42	26.3
5/L	0	.05	28.04	9.04	3.54	22.90	60.6
- Decentralized (Simple OR Fusion) -							
.25/L	.60	.08	1.93	7.08	2.06	2.08	9.4
.5/L	.29	.27	1.08	6.06	1.91	6.42	26.3
5/L	0	8.86	6.31	6.73	2.44	22.90	60.6
- Centralized -							
.25/L	.58	.02	.52	6.48	1.92	3.20	7.3
.5/L	.34	.05	1.04	5.29	1.70	10.93	20.1
5/L	0	2.96	5.82	5.77	2.15	57.4	53.5
- Distributed -							
.25/L	.57	.02	.53	6.52	1.93	3.26	11.38
.5/L	.34	.06	1.04	5.29	1.70	11.82	32.2
5/L	0	2.89	5.87	5.78	2.11	57.24	83.4

(*) Normalized by the standard deviation of the position measurement error.

(**) Normalized by the standard deviation of the velocity measurement error.

(***) Average execution time for one run.

Table 6-10 Sensitivity to Position Measurement Error

σ_u	P_{PA}	N_{FT}	N_{MT}	σ_{pos}^*	σ_{vel}^{**}	T_E^{***}	N_{TR}
- Decentralized (Simple AND Fusion) -							
.0001 L	.09	0	2.98	7.21	2.73	6.18	26.3
.001 L	.09	0	2.97	8.91	3.53	6.42	26.3
.01 L	.10	.59	2.77	10.59	1.62	6.14	26.0
- Decentralized (Simple OR Fusion) -							
.0001 L	.30	.23	1.02	6.15	1.88	6.18	26.3
.001 L	.29	.27	1.08	6.06	1.91	6.42	26.3
.01 L	.20	.59	1.10	10.59	1.62	6.14	26.0
- Centralized -							
.0001 L	.37	.02	1.00	5.22	1.67	10.36	20.1
.001 L	.34	.05	1.04	5.29	1.70	10.93	20.1
.01 L	.29	.23	1.04	8.32	1.84	10.12	19.7
- Distributed -							
.0001 L	.38	.02	1.00	5.22	1.67	11.65	32.2
.001 L	.34	.06	1.04	5.29	1.70	11.82	32.2
.01 L	.29	.21	1.04	8.31	1.84	11.27	31.6

(*) Normalized by the standard deviation of the position measurement error.

(**) Normalized by the standard deviation of the velocity measurement error.

(***) Average execution time for one run.

Table 6-11 Sensitivity to Velocity Measurement Error

σ_v	P_{PA}	N_{FT}	N_{MT}	σ_{pos}^{*}	σ_{vel}^{**}	T_E^{***}	N_{TR}
- Decentralized (Simple AND Fusion) -							
.0001 L/Δt	.09	0	2.96	9.59	2.83	5.94	26.4
.001 L/Δt	.09	0	2.97	8.91	3.53	6.42	26.3
.01 L/Δt	.08	0	2.99	9.71	2.28	6.74	27.5
- Decentralized (Simple OR Fusion) -							
.0001 L/Δt	.30	.21	1.03	5.99	1.86	5.94	26.4
.001 L/Δt	.29	.27	1.08	6.06	1.91	6.42	26.3
.01 L/Δt	.14	.95	1.32	7.56	2.18	6.74	27.5
- Centralized -							
.0001 L/Δt	.36	.03	1.01	5.04	1.61	10.24	20.1
.001 L/Δt	.34	.05	1.04	5.29	1.70	10.93	20.1
.01 L/Δt	.27	.38	1.22	5.94	1.96	11.96	21.3
- Distributed -							
.0001 L/Δt	.36	.04	1.02	5.06	1.60	11.32	33.7
.001 L/Δt	.34	.06	1.04	5.29	1.70	11.82	32.2
.01 L/Δt	.23	.39	1.26	6.21	2.05	18.18	33.6

(*) Normalized by the standard deviation of the position measurement error.

(**) Normalized by the standard deviation of the velocity measurement error.

(***) Average execution time for one run.

Table 6-12 Sensitivity to False Alarm Rate

ν_{FA}	P_{PA}	N_{FT}	N_{MT}	σ^{*}_{pos}	σ^{**}_{vel}	T^{***}_{E}	N_{TR}
- Decentralized (Simple AND Fusion) -							
2	.06	0	2.73	9.37	2.72	3.18	16.5
4	.09	0	2.97	8.91	3.53	6.42	26.3
8	.08	0	2.69	12.87	3.69	5.38	19.4
- Decentralized (Simple OR Fusion) -							
2	.32	.21	1.02	6.34	2.02	3.18	16.5
4	.29	.27	1.08	6.06	1.91	6.42	26.3
8	.25	1.26	1.26	6.94	2.10	5.38	19.4
- Centralized -							
2	.38	.04	.98	5.17	1.77	5.20	12.8
4	.34	.05	1.04	5.29	1.70	10.93	20.1
8	.22	.12	1.37	6.11	1.98	8.79	14.9
- Distributed -							
2	.38	.04	.98	5.17	1.77	6.27	20.6
4	.34	.06	1.04	5.29	1.70	11.82	32.2
8	.22	. 12	1.40	6.17	1.99	8.57	23.4

(*) Normalized by the standard deviation of the position measurement error.
(**) Normalized by the standard deviation of the velocity measurement error.
(***) Average execution time for one run.

Table 6-13 Sensitivity to Probability of Detection

P_{Dmax}	P_{PA}	N_{FT}	N_{MT}	σ_{pos}^{*}	σ_{vel}^{**}	T_{E}^{***}	N_{TR}
- Decentralized (Simple AND Fusion) -							
.5	.10	0	2.57	8.79	1.91	7.50	30.8
.7	.09	0	2.97	8.91	3.53	6.42	26.3
.9	.09	0	2.82	9.70	2.08	2.55	10.8
- Decentralized (Simple OR Fusion) -							
.5	.21	.16	1.37	7.02	2.03	7.50	30.8
.7	.29	.27	1.08	6.06	1.91	6.42	26.3
.9	.46	.29	.56	6.47	1.94	2.55	10.8
- Centralized -							
.5	.23	.17	1.34	5.60	1.85	13.61	24.4
.7	.34	.05	1.04	5.29	1.70	10.93	20.1
.9	.51	.09	.63	4.89	1.71	3.97	8.39
- Distributed -							
.5	.24	.12	1.36	5.64	1.86	15.3	38.8
.7	.34	.06	1.04	5.29	1.70	11.82	32.2
.9	.51	.09	.63	4.89	1.71	11.89	16.8

(*) Normalized by the standard deviation of the position measurement error.

(**) Normalized by the standard deviation of the velocity measurement error.

(***) Average execution time for one run.

6.3.4.1 Pruning Threshold

Table 6-8 examines the effect of the pruning threshold. The centralized and distributed schemes perform quite equally and show the same tendencies. With a low pruning threshold, both schemes perform better but require more computational resources, and with a higher threshold, perform worse but with less resources. On the contrary, it is more difficult to explain how the two decentralized schemes′ performance varies with the pruning threshold. The worse performance under high pruning threshold is naturally expected. However, the two decentralized schemes also perform worse than the baseline with the lower pruning threshold. This result is probably due to the failure of simple fusion schemes used in the two decentralized schemes, or the infrequent communication which tends to produce very different situation assessments among the two nodes, or a combination of these two factors. As far as the computational requirements are concerned, the two decentralized schemes behave normally.

6.3.4.2 Target Density

The response to the change in the target density, shown in Table 6-9, also displays a similar trend. When the target density is high, we see a clear difference in performance between the two groups of schemes, i.e., (1) the two decentralized schemes and (2) the centralized and distributed schemes. When the target density is low, however, the difference in performance becomes less obvious. Even the two decentralized schemes perform well since the chance of missing targets is very low.

On the other hand, with a high target density, the two decentralized schemes´ performance obviously deteriorates. As in the previous comparison, the responses of the centralized and the distributed schemes are very similar.

6.3.4.3 Other Parameters

The effect of measurement errors, shown in Tables 6-10 and 6-11, shows more or less expected general trends. Namely, the larger the measurement error, the more confusing the measurements become and the more the performance of all of the schemes degrades. However, the response of the two decentralized schemes is rather flat compared with the centralized and distributed scheme. This indicates a certain degree of robustness of the relatively simple decentralized schemes and their inability to take advantage of the improved external condition (due to the infrequent communication and use of a rather heuristic fusion scheme). In the last two comparisons, shown in Table 6-12 (false alarm rate) and Table 6-13 (probability of detection), we see generally the same trend. However, as we have observed in other comparative statistics, the performance of the two decentralized schemes does not change much. In particular, the simple AND-logic responds to the external conditions in a direction opposite to the expected direction.

6.4 SUMMARY OF NUMERICAL EXAMPLES

This section summarizes the results obtained in the two numerical examples. The two simple examples were chosen so that we could isolate the basic characteristics of each of the three communication schemes more readily. These examples, however, represent two radically different situations.

Example 1:

The targets were stationary and both nodes had the same field-of-view. Thus, there was a high degree of informational overlap (i.e., redundancy), and each node could have performed tracking adequately by itself under normal conditions. This situation results in fairly respectable performance for the decentralized scheme (no communication). On the other hand, since the data from the two nodes are highly correlated (being from the same set of stationary targets), the performance is sensitive to the frequency and type of communication (decentralized versus centralized versus distributed). Furthermore, since correlated data implies the storage and evaluation of a large number of hypotheses, any change in a key parameter such as the pruning threshold affects the performance differently for different schemes.

Example 2:

On the contrary, in Example 2, the targets were moving in one direction and the fields of view of the sensors in the two nodes were disjoint. therefore, the two nodes received mostly independent and complementary information. In such a case, communication becomes essential and can improve the performance substantially. Particularly, when each node is operating independently (distributed or decentralized scheme), proper coordination (both communication and information fusion) becomes more important. Moreover, since targets were moving, the timing of communication would be quite critical. For example, in a distributed scheme, a set of measurements may be erroneously dismissed as false alarms if it is obtained right after a communication instant; the node would have to wait a long time before the next round of communication. The same set of measurements could have been confirmed correctly to have come from a target if the other node could communicate the presence of a track at a crucial time.

In short, Example 1 is a case where the centralized scheme is vulnerable because of the large amount of correlated data, while Example 2 is one where the distributed scheme is vulnerable. As a result, we have observed a slight advantage of the distributed scheme over the centralized scheme in Example 1, and almost identical performance in Example 2. When we consider the non-zero pruning threshold as a proxy of the computational resources, the results of Example 1 clearly show an advantage of distributed processing. The external environment in Example 2 is generally unfavorable to the distributea scheme. However, the results indicate that with proper coordination such as the information fusion algorithm used in the distributed GTC, the same performance as the centralized scheme can be achieved.

From both examples, we notice the quality of information is as important as the quantity of information. We have found that when the quality of information is low (e.g., when the false alarm rate is high), the quantity of information does not positively correlate with the performance. This observation is, to some degree, contrary to the results of conventional filtering systems where it is always beneficial to have more data.

It is difficult to draw general conclusions from the simulation experiments since performance depends on environmental factors (target density, false alarm rates, measurement error) as well as computing resources (pruning threshold). Based on the simulation results, however, we can draw the following conclusions. When the amount of data is large and the quality is low (high target density, low detection problity, and high false alarm rates), a distributed scheme where only hypotheses are communicated is generally prefered since the amount of data handled at each node is smaller. The advantage can only be realized, however, if proper coordination in terms of communication times and fusion algorithms is used. An ad hoc coordination algorithm, such as that used in the decentralized schemes in Example 2, may not perform well. These advantages are in addition to others such as reliabiliy, cost, etc. mentioned in the introduction of this report.

7. CONCLUSIONS

In this report, we have described the results of our research on the distributed situation assessment problem using a distributed sensor network. We had two specific goals for our research:

- investigate techniques of hypothesis representation, formation, evaluation, etc., in a distributed sensor network;
- investigate various tradeoffs such as computation versus communication, and the performance of centralized, decentralized and distributed structures as a function of various parameters.

Although we dealt mostly with general but highly idealized models, the tracking and classification of multiple targets in a low signal-to-noise ratio and high cluttered environment was chosen as an application area to focus our attention. Our approach had been both analytical and heuristic. Exact algorithms were developed using precise mathematical models and combined with more heuristic rules in their implementation. Simulation experiments were also conducted to understand issues which are not amenable to analytic studies.

To provide a mathematical foundation for the multitarget tracking problem, we have developed a theory for multitarget tracking and classification. The centralized version has been implemented in the form of the Generalized Tracker/Classifier, which includes many existing trackers as special cases. This theory addresses the issues of how hypotheses should be represented, formed, evaluated and managed in the processing of local data.

The centralized algorithm was decomposed to obtain the Distributed Generalized Tracker/Classifier. The processing architecture at each node was specified and consists of the following three modules: the Generalized Tracker/Classifier for processing of local sensor data, the information fusion module, and the information distribution module. We have thus addressed many of the issues associated with information integration in a network. The general problem of distributed estimation by a network of agents has also been considered. Algorithms which allow each agent to integrate or fuse the information from other agents without redundant use of the same information have been devised.

The algorithms have been tested on two different scenarios to evaluate their performance. The sensitivities of the performance of various communication schemes to several parameters were investigated. We found that having more data, as in a centralized situation, is not necessarily better unless resources are available to process the data. In general, the quality of the information is more important than the amount of data. With a properly coordinated distributed scheme, where only hypotheses are communicated, performance similar to that of the centralized scheme can be achieved.

Our research has addressed some of the basic issues related to the design and operation of a DSN. Specifically, we have developed algorithms for distributed multitarget tracking and classification, investigated their performance and compared it with other communication schemes. To fully capitalize on the potential of a DSN, we need to address some other issues. These include:

- how to handle large networks with many heterogeneous sensors
- how to schedule the communication among the nodes efficiently
- how to allocate the sensor resources to optimize the performance of the network
- how to make the nodes adapt to changing network conditions such as failures of nodes and communication links
- how to evaluate the performance of such a distributed system
- how to reduce the vulnerability of the DSN to hostile activities

Some of these issues can be addressed mathematically, while others have to be handled by more heuristic or symbolic techniques such as artificial intelligence. In addition, the real time implementation of these algorithms in a node also poses some very relevant problems in computer (VLSI) architecture design.

REFERENCES

1. C.Y. Chong, S. Mori, E. Tse and R.P. Wishner, "A General Theory for Bayesian Multitarget Tracking and Classification - Generalized Tracker/Classifier (GTC)," AI&DS Technical Report TR-1505-1, December, 1982.

2. S. Mori, C.Y. Chong, R.P. Wishner and E. Tse, "Multitarget Multisensor Tracking Problems: A General Bayesian Approach," Proc. 1983 American Control Conference, San Francisco, CA, 1983.

3. C.Y. Chong, S. Mori, E. Tse and R.P. Wishner, "Distributed Estimation in Distributed Sensor Networks," Proceedings of 1982 American Control Conference, Arlington, VA, 1982.

4. C.Y. Chong, E. Tse, and S. Mori, "Distributed Estimation in Networks," Proceedings of 1983 American Control Conference, San Francisco, CA. 1983.

5. D.B. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. Automat. Contr., Vol. AC-24, No. 6, pp. 843-854, Dec. 1979.

6. K.M. Keverian and N.R. Sandell, Jr., "Multi-object Tracking by Adaptive Hypothesis Testing," Report LIDS-R-959, MIT, December, 1979.

7. H.W. Sorenson, and D.L. Alspach, "Recursive Bayesian Estimation Using Gaussian Sums," Automatica, Vol. 7, No. 4, pp. 465-479, July, 1971.

8. Y. Bar-Shalom, "Extension of the Probabilistic Data Association Filter to Multitarget Environment," Proc. 5th Symp. Nonlinear Estimation, Univ. Calif., San Diego, Sept. 1974.

9. T.E. Fortmann, Y. Bar-Shalom and M. Scheffe, "Multitarget Tracking Using Joint Probabilistic Data Association," Proc. 1980 IEEE Conf. on Decision and Control, Albuquerque, NM, Dec. 1980.

10. D.L. Alspach, "A Gaussian Sum Approach to the Multitarget Identification-Tracking Problem," Automatica, Vol. 11, pp. 285-295, May 1975.

11. R.A. Singer, R.G. Sea and K. Housewright, "Deviation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 423-432, July 1974.

12. C.L. Morefield, "Application of 0-1 Integer Programming to

Multitarget Tracking Problems," IEEE Trans. Automat. Contr., Vol. AC-22, pp. 302-312, June 1977.

13. F. Bourgeois and J.-C. Lassalle, "An Extension of the Munkres Algorithm for the Asignment Problem to Rectangular Matrices," Communications of ACM, Vol. 14, No. 12, pp. 802-804, Dec. 1971.

APPENDIX A

MULTITARGET MULTISENSOR TRACKING PROBLEM - PART 1: A GENERAL SOLUTION AND A UNIFIED VIEW ON BAYESIAN APPROACHES

S. Mori, C.Y. Chong, E. Tse, and R.P. Wishner

ABSTRACT

Based upon a general target/sensor model, a very general solution to the multitarget tracking problem is derived. When this solution is applied to a special class of models consisting of independent, identically distrbuted (i.i.d.) target models, a less general but more implementable class of multitarget tracking algorithms is obtained. Some existing algorithms are then examined based upon a unified view created by our derivation of general tracking algorithms. Part 1 covers most of the analytic results, while in Part 2, hypothesis management and other issues pertaining to implementation of multitarget tracking algorithms are discussed with a simple numerical example.

I. Introduction

During the past decade, the multitarget tracking problem has attracted a great number of researchers, especially in the fields of control and estimation. The problem is both theoretically interesting and very important in terms of applications. Technically, it calls for a new body of theory or a large collection of standard techniques from various fields such as modelling, stochastic inference, nonlinear filtering, etc. Its wide range of applications includes anti-missile/aircraft defense, air traffic control, ocean/battlefield surveillance, etc. Past achievements in this area are well documented in the survey paper by Bar-Shalom [1] and the Naval Ocean Surveillance Correlation Handbooks, [2] and [3]. The introductory section of the paper by Reid [4] also contains a short but excellent survey.

Despite many efforts in this area, the present stage of research may well be characterized as an unorganized collection of numerous "named" or "unnamed" algorithms. An attempt to create a unified view of these algorithms is described in a recently published paper [5]. However, the focus is on the relationship between multitarget tracking and other new topics such as event-driven linear systems, etc., and on the interpretation of Reid's algorithms described in [4]. The object of our paper is to provide a general Bayesian solution to a general but mathematically rigorous model and to provide a unified view of Bayesian approaches to the multitarget tracking problem. In doing so, we may have clearer interpretations of many existing algorithms and a better understanding of what is necessary for future theoretical developments in this area.

In short, the multitarget tracking problem is concerned with tracking an unknown number of targets using noisy measurements whose origins are not certain and which may not originate from any target at all (false alarms, clutters, etc.). The basic and crucial deviation from conventional estimation problems is the fact that targets (objects to be tracked) as well as measurements (returns, sensor outputs) are modeled properly only when they are considered as random sets in the sense defined in [6]. Namely, (1) the number of targets, the number of measurements, etc., are random and (2) the targets and the measurements are essentially unordered tuples. For example, targets do not have a priori labels and the measurement tuple (a,b) has the same meaning as (b,a). We may (tentatively) call such a nature a random-set property or feature. In other words, one of the fundamental features of multitarget tracking is the random-set feature. Thus the uncertainty of origins of the measurements data is naturally modeled as a stochastic system which converts a random set (the set of targets) into many other random sets (the measurement data sets).

Theories of random sets are mainly concerned with uncountable-set-valued random sets and are mathematically highly sophisticated. Fortunately, when we restrict ourselves to random sets whose cardinalities are finite with probability one, we can still apply standard probabilistic techniques. For example, a random finite set X of reals can be probabilistically completely described by specifying a probability Prob.$\{\#(X)=n\}$ (In this paper, $\#(A)$ is the cardinality of a set A.) for each nonnegative integer n and a joint probability distribution density function $p_n(x_1,\ldots,x_n)$ of elements of the set for each positive n. In order for this specification to be appropriate, we must require every p_n to be interchangeable (permutable). This is the basic approach which we take in this paper. As in almost all the existing literature on multitarget tracking, the basic task is to hypothesize the origin of each measurement and to evaluate every possible hypothesis, or in other words, to create and rank all the possible combinations. Although one may discern some similarity between multitarget tracking algorithms and classical hypothesis testing formula, especially chi-square testing, the difference in underlying models is very obvious. To achieve this basic task, we propose new definitions for tracks and hypotheses, which we believe are both mathematically rigorous and intuitively appealing, and in fact are included, at least implicitly, in almost all the existing multitarget tracking algorithms.

In many cases, in order to broaden one's perspective and obtain deeper understanding, it is best to start with a general model and go into greater detail later with a more restrictive class of models. In the rest of this paper we will proceed according to this philosophy, defining a fairly general model in the next section and following that with two sections in which the definition and the Bayesian evaluation of hypotheses are described. Then we will discuss an important subclass of problems, i.e., what we may call i.i.d. (independent, identically distributed) target models. The importance of this subclass is two-fold: (1) It provides us with a set of implementationally feasible algorithms; (2) A unified view of existing algorithms will emerge. Part 1 covers most of the theoretical issues whereas Part 2 describes hypothesis management and other implementation issues with a simple numerical example.

II. Target and Sensor Models

A. Target Model

In our terminology, a target is a generic name for the smallest unit of object which, when detected by a sensor at a certain time, generates some measurement(s) in the sensor's output with a certain probability. In our general model, all targets of interest are modeled as one entity rather than as a collection of individual targets. Formally, a target system state at time t is a realization $(X(t),N_T(t))$ at time t of a continuous-time stochastic process (X,N_T) on a target system state space which is the direct-sum space

$$\bigcup_{n=0}^{\infty} \mathcal{X}_n \times \{n\}$$

of a system $\{\mathcal{X}_n\}$ of hybrid sets. By a hybrid set, in this paper, we mean a direct product space of a subset of a Euclidean space (called continuous part) and a finite set (called discrete part). The use of hybrid sets allows us to consider different kinds (types) of targets, sudden structural changes in dynamics (maneuvering targets), changes in operational modes, etc., as well as the usual physical states such as positions and velocities. The second element, N_T, represents the total number of targets in the system. When n=0, $\mathcal{X}_n$ is defined as $\{\theta\}$ where θ is merely a symbol for "no target" and $\theta \notin \mathcal{X}_n$ for all n. For this paper, we make the following assumption:

Assumption 1: (Constant Number of Targets and Markovian Property)

The component N_T of the target system state (X,N_T) is a constant but random nonnegative integer with a given probability distribution. For each positive integer n, given $N_T=n$, X is a time-homogeneous Markov process on $\mathcal{X}_n$ associated with an initial distribution

$$Q_o^n(dX) = \text{Prob.}\{X(t_0)\varepsilon dX|N_T=n\} \tag{1}$$

for a fixed t_0 and a transition probability

$$F_{\Delta t}^n(dX|X) = \text{Prob.}\{X(t+\Delta t)\varepsilon dX|X(t)=X,\ N_T=n\} \tag{2}$$

for each $(X,\Delta t)\varepsilon\mathcal{X}_n\times[0,\infty)$ and each $t\varepsilon[t_0,\infty)$.

The time-homogeneity (stationary transition) assumption can be easily removed but helps the notational complexity in this paper. The requirement for N_T to be constant is not very restrictive. For example, to consider the possibility of disappearing targets we may include a component such as $\{$active, inactive$\}$ in the target system state space and construct an appropriate birth-death-type Markov process.

For each positive n, we assume that the component $\mathcal{X}_n$ of the target system state space is further decomposed into two parts as

$$\mathcal{X}_n = \mathcal{X}_n^C\times\mathcal{X}_n^I$$

where $\mathcal{X}_n^C$ is the space for the part representing the <u>common target state</u> and $\mathcal{X}_n^I = \underbrace{\mathcal{X}_n^i\times\cdots\cdots\times\mathcal{X}_n^i}_{n}$ is the direct product of n identical <u>individual target state spaces</u>, $\mathcal{X}_n^i$. A simple example is the one in which $\mathcal{X}_n^C=\{0\}$ (no common state space) and $\mathcal{X}_n^i$ is a Euclidian space. Another example is a model with $\mathcal{X}_n^C = \mathcal{X}_n^i = R^2$ (the set of pairs of reals), in which $(x_0, x_1, \ldots, x_n) \varepsilon \mathcal{X}_n$ represents a target system state for a group with a hypothetical centroid x_0 and x_i being the deviation of the i-th target from x_0. The inclusion of such a component $\mathcal{X}_n^I$ is necessary in order to model the random-set nature of targets. For this purpose, we must require a priori interchangeability of individual targets as precisely defined in Assumption 2 below. In the rest of this paper, we call a function $\Pi: \mathcal{X}_n \to \mathcal{X}_n$ a n-target permutation homeomorphism (induced by permutation π) if, for every $(X_C, (X_i)_{i=1}^n) \varepsilon \mathcal{X}_n = \mathcal{X}_n^C \times \mathcal{X}_n^I$,

$$\Pi(X_C, (X_i)_{i=1}^n) = (X_C, (X_{\pi(i)})_{i=1}^n) \qquad (3)$$

with a permutation π on $\{1, \ldots, n\}$.

Assumption 2: (Interchangeability (1))

For each positive n, Q_0^n and $F_{\Delta t}^n$ are interchangeable (or permutable) with respect to the individual target state part $\mathcal{X}_n^I$ of $\mathcal{X}_n$, i.e.,

$$Q_0^n(\Pi(dX)) = Q_0^n(dX) \qquad (4)$$

and

$$F_{\Delta t}^n(\Pi(dX)|\Pi(X)) = F_{\Delta t}^n(dX|X) \quad , \qquad (5)$$

for any n-target permutation homeomorphism Π on $\mathcal{X}_n$.

Assumption 2 assures us that we can use any of the equi-probable target system distributions caused by permutation of targets. In other words, under

Assumption 2, we can choose any one of the permutations and assume that the order of the targets is given in that way. Under Assumptions 1 and 2, we can construct a wide range of target models, including those in which targets move in a group rather than individually, i.e., their motions are not independent but correlated.

B. Sensor Model

Let S be a finite set of sensors in the system. In this paper, each sensor $s \in S$ is modeled as a generic mechanism which observes the target system state space and generates a finite set of measurements, called a data set, intermittently according to a certain sampling pattern. Each measurement in a data set from a sensor $s \in S$ is an element of the measurement value space $\mathcal{Y}_s$ which is a hybrid space with the direct-product measure μ_s of the Lesbegue measure on the continuous part and the counting measure on the discrete part. The continuous part of $\mathcal{Y}_s$ is used for analog information such as positions and velocities whereas the discrete part is for feature-type information such as size/cross-section classification of aircraft radar images, track/wheel classification of ground vehicle images, etc.

Formally a data set is a random element $((y_j)_{j=1}^{N_M}, N_M, t, s)$ in the data set space

$$\bigcup_{m=0}^{\infty} \bigcup_{s \in S} (\mathcal{Y}_s)^m \times \{m\} \times [t_0, \infty) \times \{s\}.$$

where $(\mathcal{Y}_s)^m = \underbrace{\mathcal{Y}_s \times \ldots . \times \mathcal{Y}_s}_{m}$ when $m>0$ and $(\mathcal{Y}_s)^0 = \{\theta\}$ (θ is a symbol for "no meament.") A quadruple $((y_j)_{j=1}^{N_M}, N_M, t, s)$ in this space is interpreted as a data set generated by sensor s at time t and containing N_M measurements, $y_1,\ldots,y_{N_M}$. In our generic sensor model, the generation of data sets by sensors is modeled by a four-step mechanism: (1) detection, (2) number-of-false-alarm generation, (3) random assignment and (4) measurement value generation. First we assume a certain sensor scheduling mechanism which determines what sensor is activated when. Once a sensor $s \varepsilon S$ is activated at time t a data set $((y_j(t,s))_{j=1}^{N_M(t,s)}$, $N_M(t,s)$, t, s) is generated instantaneously through the following mechanisms:

(1) Detection:

A detected target set is a unit which generates one measurement in the sensor's output. Such a set is modeled by a detected target set collection which is a random collection $D(t,s)$ of nonempty subsets of positive integers such as

$$\text{Prob.}\{\bigcup D(t,s) \subseteq I_T | N_T\} = 1 \ , \tag{6}$$

where

$$I_T = \{1,\ldots.,N_T\} \quad . \tag{7}$$

$\{i_1,i_2\} \varepsilon D(t,s)$ means that the i_1-th and the i_2-th targets are detected and create one measurement in the data set. Thus $\bigcup D(t,s)$ is the set of all the detected targets. The random nonnegative integer defined by

$$N_D(t,s) = \#(D(t,s)) \tag{8}$$

is called the number of detected target sets.

(2) Number of False Alarms

A measurement in a data set is called a false alarm (a generic name for clutter, false return, etc.) if it does not originate from any target. In our model, the origins of non-false-alarm measurements are D(t,s) and the number of false alarm measurements is represented by a random nonneagative integer $N_{FA}(t,s)$. Thus the random nonnegative integer $N_M(t,s)$ is determined by

$$N_M(t,s) = N_D(t,s) + N_{FA}(t,s) \ . \tag{9}$$

(3) Random Assignment

Each of the $N_M(t,s)$ measurements in the data set originates from one of the $N_D(t,s)$ detected target sets or is one of the $N_{FA}(t,s)$ false alarms. Each origin is determined by a random assignment A(t,s) which is a one-to-one integer-valued random function such that

$$\text{Prob.}\{\text{Dom}(A(t,s))=D(t,s),\text{Image}(A(t,s))\subseteq J_M(t,s)|N_M(t,s)\} = 1 \tag{10}$$

where

$$J_M(t,s) = \{1,....,N_M(t,s)\}, \tag{11}$$

and Dom(f) and Image(f) are the domain and the image of a function f. Define a random set $J_{FA}(t,s)$ by

$$J_{FA}(t,s) = J_M(t,s)\setminus\text{Image}(A(t,s)). \tag{12}$$

$d\varepsilon D(t,s)$ and $A(t,s)(d)=j$ means that the j-th measurement originates from a detected target set d, whereas $j\varepsilon J_{FA}(t,s)$ means the j-th measurement is a false alarm.

(4) Measurement Values

Finally, given the number of measurements, $N_M(t,s)$, and the origin of each j in $J_M(t,s)$, the measurement value vector $(y_j(t,s))_{j=1}^{N_M(t,s)}$ is generated, completing the data-set-generating mechanism.

For the rest of this paper, the following five assumptions are made:

Assumption 3: (Known Exact Timing)

Every sensor generates a finite number of data sets within any finite time interval. The time at which any data set is generated is exactly known and completely determined by each individual sensor (not by any other factor correlated with the target system state).

Assumption 4: (Memory-less Sensor)

There is no memory in any sensor, so that any single-data-set statistics conditioned on the current target system state and any other statistical condition are the same as the ones conditioned only on the current target system state.

Assumption 5: (No Merged Measurements)

No measurement in any data set from any sensor originates from two of more targets.

Assumption 6: (No Split Measurements)

No target generates more than one measurement in any data set.

Assumption 7: (Random Order)

The order of measurements in any data set contains no information about the target system state.

Assumptions 3 and 4 are standard in filtering problems and allow us to use standard techniques used in sampled-data or discrete-time filtering problems. Assumptions 5 and 6 imply that, for any $(t,s)\varepsilon[t_0,\infty)\times S$, $D(t,s)$ is a disjoint collection of singletons so that we can define a random set $I_D(t,s)$ by

$$I_D(t,s) = \{i|\{i\}\varepsilon D(t,s)\} \tag{13}$$

such that $\text{Prob.}\{I_D(t,s)\subseteq I_T|N_T\}=1$ and a binary random function $F_D(t,s)$ by

$$F_D(t,s) = \chi(i\ ;\ I_D(t,s)) \tag{14}$$

where $\chi(\cdot;A)$ is the indicator function of a set A. The random function $F_D(t,s)$ is called the <u>detection function</u>. Then, due to Assumptions 3 and 4, the detection mechanism is completely modeled by specifying the <u>detection probability function</u>

$$P_D(\delta|X,n,t,s) = \text{Prob.}\{F_D(t,s)=\delta|X(t)=X,N_T=n\} \tag{15}$$

for every $(\delta,X,n,t,s)\varepsilon\bigcup_{n=0}^{\infty}\mathcal{D}(n)\times\mathcal{X}_n\times\{n\}\times[t_0,\infty)\times S$ such that $\sum_{\delta\varepsilon\mathcal{D}(n)}P_D(\delta|X,n,t,s)=1$ for any positive n, where $\mathcal{D}(n)$ is the set of all the binary function defined on $\{1,\ldots,n\}$. The same couple of assumptions allows us to describe the number-of-false-alarms generation by specifying the <u>number-of-false-alarms probability function</u>

$$P_{N_{FA}}(m|\delta,X,n,t,s) = \text{Prob.}\{N_{FA}(t,s)=m|F_D(t,s)=\delta,X(t)=X,N_T=n\}\ . \tag{16}$$

Let $\mathcal{A}^o(I,J)$ be the set of all the one-to-one function defined on I taking values in J. Then we have

$$\text{Prob.}\{A(t,s))\varepsilon\mathcal{A}^o(D(t,s),J_M(t,s))\,|D(t,s),J_M(t,s)\} = 1 \tag{17}$$

and Assumption 7 implies

$$\text{Prob.}\{A(t,s)=\alpha \mid D(t,s),J_M(t,s)\} = \text{Prob.}\{A(t,s)=\alpha' |D(t,s),J_M(t,s)\} \tag{18}$$

for all pairs (α,α') of elements in $\mathcal{A}^o(D(t,s),J_M(t,s))$. Hence we have

$$\text{Prob.}\{A(t,s)=\alpha|N_M(t,s),D(t,s),X(t),N_T\} = \frac{(N_M(t,s) - N_D(t,s))\ !}{(N_M(t,s))\ !} \tag{19}$$

for each α in $\mathcal{A}^o(D(t,s),J_M(t,s))$. Finally the sensor model is completed by specifying the <u>measurement value probability density function</u> P_M defined by

$$P_M(y|\alpha,m,\delta,X,n,t,s)\mu_s^m(dy) = \text{Prob.}\{y\varepsilon dy|A(t,s)=\alpha,N_M(t,s)=m,F_D(t,s)=\delta,X(t)=X,N_T=n\} \tag{20}$$

for every $(y,\alpha,m,\delta,X,n,t,s)$ where μ_s^m is the m-tuple direct-product measure of μ_s.

It is clear that our general sensor model as well as our general target model allows us to consider a variety of modern sensor systems. One should particularly note that the probability of detection is generally dependent on the target system state. Therefore, the absence of returns is at least potentially as informative as their presence. For example, for a sensor monitoring the radio communication of target(s), the probability of detection is zero when the equipment is shut off, and the on/off of such equipment should be included in the target system state. In particular, N_M=0 or no measurement is a data set in our model and considered a potential piece of information. It should also be noted that in our general sensor model every data set contains "number-of-measurements" information, and hence, every sensor is a type 1 sensor in Reid's terminology in [4]. A type 2 sensor in his terminology is a sensor which creates data sets with N_M(number-of-measurement)=0 or 1 with probability one in our model and is not (at least in principle) treated separately. Of course, Assumption 3 has a crucial role in such a treatment as ours.

For sensor systems which involve measurement time delays dependent on the target state (e.g., acoustic sensor systems described in [7]), a straightforward model in which the target state is a pair (position,velocity) and the sensor measurements are ranges and/or bearings violates Assumptions 3 and 4 In such a case, in order for our formulation to be applicable, careful modeling is called for so that our assumptions are valid at least in an approximate sense. On the other hand, although Assumptions 5 and 6 are quite standard in multitarget literature, they are not essential to the development in this paper. Recently, an attempt was made to relax Assumption 5 to deal with merged measurements in [8]. We make these assumptions in this paper largely to minimize nonessential complexity. Another way to state the last assumption, Assumption 7, is that a data set is the smallest unit of sensor data in which the order of measurements does not contain any information about the target system state. For example, the measurements from a radar with a fixed scanning pattern may result in the order of measurements containing information about the targets. In such a case, the data sets should be further divided so that the measurement order does not contain any significant information.

Our general sensor model is completely described by specifying the detection probability function, P_D, the number-of-false-alarms probability function, $P_{N_{FA}}$, and the measurement value probability density function P_M. Finally, since our sensor model is a mechanism in which a random set generates other random sets, we need one more assumption on the target interchangeability (permutability) corresponding to Assumption 2:

Assumption 8: (Interchangeability (2))

P_D, $P_{N_{FA}}$ and P_M are invariant under the permutation of targets, i.e., $P_D(\delta\circ\pi|\Pi(X),n,t,s)$, $P_{N_{FA}}(m|\delta\circ\pi,\Pi(X),n,t,s)$ and $P_M(y|\alpha\circ\bar{\pi},m,\delta\circ\pi,\Pi(X),n,t,s)$ are all invariant with respect any n-target permutation homeomorphism Π induced by any permutation π where $\bar{\pi}(\{i\})=\{\pi(i)\}$ and $\circ$ is the function composition operation ($f\circ g(x)=f(g(x))$ for all x in Dom(f)).

III. Tracks and Hypotheses

Tracks and hypotheses are among the most frequently used terms in the multitarget tracking literature. Often, however, these are not precisely defined. Our definitions of tracks and hypotheses, given below, closely follow Morefield's notations in [9] but differ in one crucial aspect, namely the separation of the measurement-value information and the number-of-measurements information in each data set.

Let $\mathcal{Z}$ be the collection of all the data sets. Due to Assumption 3, $\mathcal{Z}$ is countable. Without loss of generality, we can assume that $\mathcal{Z}$ has a one-to-one correspondence to a subset K of $[t_0,\infty) \times S$ through the isomorhpism,

$$\begin{array}{ccc} (y,m,t,s) & \longleftrightarrow & (t,s) \; . \\ \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ \mathcal{Z} & & K \end{array}$$

Hence, for every k in K, we can denote the unique member (y,m,k) in $\mathcal{Z}$ by $Z(k)$. We may also call $Z(k)$ data set k. It is then natural to call K the <u>data set index set</u>. Let $\leq$ be any total order on K such that $(t,s) \prec (t',s')$ whenever $t < t'$. ($k \prec k'$ if $k \leq k'$ but $k \neq k'$.) Such an order may not be unique but its existence is obvious.

For each k in K, define the <u>cumulative data set</u> $Z^{(k)}$ up to k by

$$Z^{(k)} = \bigcup_{k' \leq k} Z(k') \qquad (21)$$

and the *cumulative measurement index set* $J_M^{(k)}$ up to k by

$$J_M^{(k)} = \bigcup_{k' \leq k} J_M(k') \times \{k'\} \quad , \tag{22}$$

where J_M is defined by (11).

Due to Assumption 3, we may treat K as a completely deterministic set whereas $Z^{(k)}$ and $J_M^{(k)}$ are random. Every (j,t,s) in $J_M^{(k)}$ indicates the j-th measurement in a data set from sensor s at time t. Then, for each k in K a *track* at k is a subset of $J_M^{(k)}$ and a *data-to-data association hypothesis* (henceforth referred to simply as *hypothesis*) at k is a (possibly empty) collection of nonempty track(s). A track τ at k is said to be *possible* if

$$\#((J_M(k') \times \{k'\}) \cap \tau) \leq 1 \tag{23}$$

for all $k' \leq k$ (Assumption 6). Let the set of all the possible tracks at k be denoted by $\mathcal{T}(k)$. A hypothesis λ at k is said to be *possible* if it is a subset of $\mathcal{T}(k) \setminus \phi$ and $\tau \cap \tau' = \phi$ for all the pairs (τ, τ') of tracks in λ such that $\tau \neq \tau'$ (Assumption 5). Denote the set of all the possible hypotheses at k by $\mathcal{H}(k)$.

Define a random set, via the random function A(k) and the random set $I_D(k)$, as

$$\Lambda = \left\{ \{ (A(k)(\{i\}), k) \mid k \in K \} \mid i \in \bigcup_{k \in K} I_D(k) \right\} \quad . \tag{24}$$

Its restriction to k is defined by

$$\Lambda_{|k} = \{\tau \cap J_M^{(k)} \mid \tau \in \Lambda\} \setminus \{\phi\} \tag{25}$$

for each $k \in K$. Then it is clear that, for a $\lambda \in \mathcal{H}(k)$, event $\{\Lambda_{|k} = \lambda\}$ should be interpreted as an incidence in which (1) there are $\#(\bigcup_{k' \preceq k} I_D(k')) = \#(\lambda)$ targets which are detected and included in at least one of the data sets prior to and including k, (2) each τ in λ corresponds to a target (which has been detected in at least one data set $k' \preceq k$) in a one-to-one fashion, (3) $(j,k') \in \tau$ means that the j-th measurement in data set k' originates from the target identified by τ, (4) $\tau \cap (J_M(k') \times \{k'\}) = \phi$ means that the target is "falsely" dismissed at k', and $J_M^{(k)} \setminus (\bigcup \lambda)$ is the set of all the false alarms up to k.

Therefore, every λ in $\mathcal{H}(k)$ is a hypothesized set of tracks which are in turn the sets of measurement indices which are hypothesized to originate from targets. The term "hypothesis" is thus suitable for use in our formulation. Assumptions 5 and 6 imply

$$\text{Prob.}\{\Lambda_{|k} \in \mathcal{H}(k) \mid J_M^{(k)}\} = \sum_{\lambda \in \mathcal{H}(k)} \text{Prob.}\{\Lambda_{|k} = \lambda \mid J_M^{(k)}\} = 1 \ . \tag{26}$$

In other words, $\mathcal{H}(k)$ is the mutually distinct and collectively exhaustive set of all the possible "explanations" of the origin of each measurement in the data sets up to k.

At this point a few words of caution are in order, because a straightforward expansion such as

$$\text{Prob.}\{X(t)\varepsilon dX, N_T=n \mid Z^{(k)}\} = \sum_{\lambda\,\varepsilon\,\mathcal{H}(k)} \sum_{n=0}^{\infty} \text{Prob.}\{X(t)\varepsilon dX \mid N_T=n, \Lambda_{|k}=\lambda, Z^{(k)}\}\cdot$$
$$\text{Prob.}\{N_T=n \mid \Lambda_{|k}=\lambda, Z^{(k)}\}\,\text{Prob.}\{\Lambda_{|k}=\lambda \mid Z^{(k)}\}$$

is in general meaningless and Prob.$\{X(t)\varepsilon dX, N_T \mid \Lambda_{|k}=\lambda, Z^{(k)}\}$ may not be a part of an appropriate set of variables which may constitute a state of multitarget tracker (information state), as we will see in the next section. Nonetheless, our primary objective is to evaluate every hypothesis λ or calculate Prob.$\{\Lambda_{|k} \mid Z^{(k)}\}$. Before closing this section, let us introduce a few notations which will be useful later.

Similar to (24), for each k in K and for each $k' \precsim k$,

$$\tau_{|k'} = \tau \cap J_M^{(k')} \tag{27}$$

in $\mathcal{T}(k')$ is called the <u>restriction</u> of τ ($\varepsilon\, \mathcal{T}(k)$) to $J_M^{(k')}$ (or simply to k'), and

$$\lambda_{|k'} = \{\tau_{|k'} \mid \tau \varepsilon \lambda\} \setminus \{\phi\} \tag{28}$$

in $\mathcal{H}(k')$ is called the <u>restriction</u> of λ ($\varepsilon\, \mathcal{H}(k)$) to $J_M^{(k')}$ (or simply to k'). When (λ', resp.) is the restriction (to some $J_M^{(k')}$) of a track $\tau \varepsilon \mathcal{T}(k)$ (a hypothesis $\lambda \varepsilon\, \mathcal{H}(k)$, resp.), τ' (λ', resp.) is called a predecessor of τ (λ, resp.). Successors are defined by the inverse relation. Then, it is obvious that, when a cumulative measurement set $J_M^{(k)}$ is given, the set of all the hypotheses up to k, $\mathcal{H}^{(k)} \underline{\Delta} \bigcup_{k' \precsim k} \mathcal{H}(k')$, and the set of all the tracks up to k, $\mathcal{T}^{(k)} \underline{\Delta} \bigcup_{k' \precsim k} \mathcal{T}(k')$, are both (partially) ordered with respect to the order determined by the predecessor/successor relation. Both of these ordered sets are arborescent, i.e., the set of predecessors of any element is totally ordered. For this reason, approaches similar to the one described in this paper are often referred to as hypothesis-tree or track-splitting methods.

IV. Recursive Bayesian Evaluation of Hypotheses

In this section, a recursive Bayesian evaluation of every hypothesis λ, namely, a recursive calculation of Prob.$\{\Lambda_{|k}=\lambda|Z^{(k)}\}$ is described. The main result is presented as Theorem 1, the proof of which is given in Appendix A. The calculation is made recursively with respect to the total order $\leq$ on K .

In this section, the symbol P will occasionally be used with a slight notational abuse. It will represent a conditional probability, a conditional probability density function or a mixture of both. Using P in this way, we can write our basic recursive equation as

$$P(\Lambda_{|k}|Z^{(k)}) = \frac{P(Z^{(k)},\Lambda_{|k}|Z^{(k')},\Lambda_{|k'})}{P(Z^{(k)}|Z^{(k')})} P(\Lambda_{|k'}|Z^{(k')}) \tag{29}$$

for each k in K which has an immediate predecessor k'. If we assume that $P(\Lambda_{|k'}|Z^{(k')})$ has already been calculated, since the denominator of the right hand side of (29) is the normalizing constant, the left hand side of (29) is given completely by calculating $P(Z^{(k)},\Lambda_{|k}|Z^{(k')},\Lambda_{|k'})$. Roughly speaking, this term can be expanded as

$$P(Z^{(k)},\Lambda_{|k}|Z^{(k')},\Lambda_{|k'}) = \sum_{N_T} P(N_T|\,\Lambda_{|k'},Z^{(k')})$$

$$\int_{\mathcal{X}_{N_T}} P(Z^{(k)},\Lambda_{|k}|X(t),N_T,\Lambda_{|k'},Z^{(k')})P(dX(t)|N_T,\Lambda_{|k'},Z^{(k')}) \tag{30}$$

with $k=(t,s)$. Therefore, assuming that $P(N_T|\Lambda_{|k'},Z^{(k')})$ and $P(dX(t)|N_T,\Lambda_{|k'},Z^{(k')})$ are provided by recursion, (30) can be calculated if we know $P(Z^{(k)},\Lambda_{|k}|X(t),N_T,\Lambda_{|k'},Z^{(k')})$ which can , in fact, be calculated using the generic sensor model described in Section II.

Before proceeding with further discussion, we make a few preparatory observations. For each k in K, define a random set,

$$I_D^{(k)} = \bigcup_{k'\preceq k} I_D(k'), \tag{31}$$

of the cumulative index set of detected targets. Then the definition (24) of Λ implies that $\#(\Lambda_{|k}) = \#(I_D^{(k)})$ with probability 1. Even when we hypothesize $N_T=n$ and $\Lambda_{|k}=\lambda$ for some $\lambda\epsilon\mathcal{H}(k)$ and some $n\geq\#(\lambda)$, the true origin in I_T of each track τ in λ is still uncertain. This uncertainty can be modeled by a random integer-valued function Ω_k such that $\text{Prob.}\{\text{Dom}(\Omega_k)=\Lambda_{|k},\text{Image}(\Omega_k)=I_D^{(k)}|\Lambda_{|k},I_D^{(k)}\}=1$ and defined by

$$\Omega_k(\tau)=i \qquad \text{if and only if} \qquad \tau = \{(A(k')(\{i\}),k')|k'\preceq k\} \quad . \tag{32}$$

Then Assumptions 2 and 8 imply with a simple recursive argument that

$$\text{Prob.}\{\Omega_k=\omega|\Lambda_{|k}=\lambda,N_T=n\} = (\#(\mathcal{W}(\lambda,n))^{-1} = \frac{(n - \#(\lambda))!}{n!} \tag{33}$$

for each $\lambda\ \epsilon\ \mathcal{H}(k)$, each $n \geq \#(\lambda)$ and each $\omega\epsilon\mathcal{W}(\lambda,n)$ where $\mathcal{W}(\lambda,n)$ is the set of all the one-to-one functions defined on λ taking values in $\{1,\ldots,n\}$.

Moreover, the same set of assumptions implies that, *for* any $k=(t,s)$ in K, any λ in $\mathscr{H}(k)$, any $n \geq \#(\lambda)$, any $\omega \varepsilon \mathcal{W}(\lambda,n)$, any permutation π on $\{1,....,n\}$, and any $\bar{\omega} \varepsilon \mathcal{W}(\lambda,n)$, we have

$$\text{Prob.}\{X(t)\varepsilon dX|\Omega_k=\omega,\Lambda_{|k}=\lambda,N_T=n\} = \text{Prob.}\{X(t)\varepsilon\Pi(dX)|\Omega_k=\bar{\omega},\Lambda_{|k}=\lambda,N_T=n\} \qquad (34)$$

if $\bar{\omega}(\tau)=\pi(\omega(\tau))$ *for* all τ in λ and $\Pi(\cdot): \mathcal{X}_n \rightarrow \mathcal{X}_n$ is the n-target permutation homeomorphism induced by the permutation π.

Since our sensor model described in Section II is based on a "fixed" order of targets, we must further hypothesize the correspondence between a hypothesis λ and its origin in I_T in order to calculate (30). For this reason,

$$\begin{aligned}
&\text{Prob.}\{X(t)\varepsilon dX|N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\} \\
&= \sum_{\omega\varepsilon\mathcal{W}(\lambda,n)} \text{Prob.}\{X(t)\varepsilon dX|\Omega_k=\omega,N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\}\ \text{Prob.}\{\Omega_k=\omega|N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\} \\
&= \frac{(n-\#(\lambda))!}{n!} \sum_{\omega\ \varepsilon\ \mathcal{W}(\lambda,n)} \text{Prob.}\{X(t)\varepsilon dX|\Omega_k=\omega,N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\} \qquad (35)
\end{aligned}$$

cannot be a part of the information state to be propagated to complete the recursion. Also, (35) is, in general, not a good candidate for the tracker output either. For example, suppose that $n=2$ and $\lambda=\{\tau\}$. Then the quantity,

$$\begin{aligned}
&\text{Prob.}\{X(t)\varepsilon dX|N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\} = \\
&\quad \tfrac{1}{2}\ \text{Prob.}\{X_1(t)\varepsilon dX_1,X_2(t)\varepsilon dX_2|\omega_1,n,\lambda,Z^{(k)}\} \\
&\quad + \tfrac{1}{2}\ \text{Prob.}\{X_1(t)\varepsilon dX_1,X_2(t)\varepsilon dX_2|\omega_2,n,\lambda,Z^{(k)}\},
\end{aligned}$$

with $\omega_1(\tau)=1$ and $\omega_2(\tau)=2$ does not make much sense. It is actually an over-aggregation of information.

As seen in the subsequent theorem and Appendix A, however, the following three functions can constitute an information state of the tracker: For each k in K, define

$$\begin{cases} P_{\mathscr{A}}^{(k)}(\lambda|Z^{(k)}) = \text{Prob.}\{\Lambda_{|k}=\lambda|Z^{(k)}\} \quad , & (36) \\ P_{N_T}^{(k)}(n|\lambda,Z^{(k)}) = \text{Prob.}\{N_T=n|\Lambda_{|k}=\lambda,Z^{(k)}\} \quad , \quad \text{and} & (37) \\ P_X^{(k)}(dX|\omega,n,\lambda,Z^{(k)}) = \text{Prob.}\{X(t)\varepsilon dX|\Omega_k=\omega,N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\} \ , & (38) \end{cases}$$

for each $\lambda\,\varepsilon\mathscr{A}^{(k)}$, each $\underline{n\geq}\#(\lambda)$ (by the definition (24) of Λ, obviously $P_{N_T}^{(k)}(n|\lambda,Z^{(k)})=0$ if $n<\#(\lambda)$.) and for <u>some</u> $\omega\varepsilon\mathscr{W}(\lambda,n)$. Because of (34), just one ω is enough for (38). Again, with a somewhat notationally abusive usage of P, we have the following Bayesian expansion:

$$P(Z^{(k)},\Lambda_{|k}|Z^{(k')},\Lambda_{|k'}) = \sum_{\Omega_k}\ \sum_{\Omega_{k'}}\ \sum_{N_T} P(Z^{(k)},\Omega_k,\Lambda_{|k}|\Omega_{k'},N_T,\Lambda_{|k'},Z^{(k')})$$

$$P(\Omega_{k'}|N_T,\Lambda_{|k'},Z^{(k')})P(N_T|\Lambda_{|k'},Z^{(k')}) \qquad (39)$$

In the first term of the right hand side of (39), there is no longer ambiguity in the origins of measurements. The rest of the terms is given by (33) and (37).

The final form of our main result is stated by the following theorem:

<u>Theorem 1:</u> Under Assumptions 1 to 8, for any $k=(t,s)$ in K with an immediate predecessor $k'=(t',s')$, when $Z(k)=(y,m,k)$ is given, we have

$$P_{\mathscr{H}}^{(k)}(\lambda|Z^{(k)}) = \frac{P_{\mathscr{H}}^{(k')}(\lambda|_{k'}|Z^{(k')})}{P_Z^{(k)}(Z^{(k)}|Z^{(k')})} \cdot \frac{(m - n_D(\lambda|k))!}{m!} \cdot \sum_{n=\#(\lambda)}^{\infty} \frac{(n - \#(\lambda|_{k'}))!}{(n - \#(\lambda))!} \; P_{N_T}^{(k')}(n|\lambda|_{k'},Z^{(k')}) \cdot \mathcal{L}(y,m,n,\lambda,k) \qquad (40)$$

for each λ in $\mathscr{H}(k)$, where $P_Z^{(k)}(Z^{(k)}|Z^{(k')})$ is the normalizing constant, $n_D(\lambda|k)$ is the number of detected targets at k which λ hypothesizes, i.e.,

$$n_D(\lambda|k) = \#(\{\tau\varepsilon\lambda|(j,k)\varepsilon\tau \text{ for some } j\}) \quad , \qquad (41)$$

$\mathcal{L}(y,m,n,\lambda,k)$ is the <u>likelihood of (y,m) given (n,λ) at k</u> and is defined by

$$\mathcal{L}(y,m,n,\lambda,k) = \int_{\mathfrak{X}_n} P_M(y|\alpha,m,\delta,X,n,k) P_{N_{FA}}(m-n_D(\lambda|k)|\delta,X,n,k) \; P_D(\delta|X,n,k) \int_{\mathfrak{X}_n} F_{\Delta t}^n(dX|X') P_X^{(k')}(dX'|\omega',n,\lambda|_{k'},Z^{(k')}) \qquad (42)$$

with $\Delta t=t-t'$, for some $\omega'\varepsilon\mathscr{W}(\lambda|_{k'},n)$, $\delta\,\varepsilon\mathscr{D}(n)$ and $\alpha\varepsilon\mathscr{A}^o(\{\{i\}|\delta(i)=1\},\{1,\ldots,m\})$ which are determined by

$$\delta(i) = \begin{cases} 0 & \text{if } i \notin \text{Image}(\omega) \\ \#(\tau \cap (\{1,\ldots,m\}\times\{k\})) & \text{if } i=\omega(\tau) \text{ and } \tau\ \varepsilon\ \lambda \end{cases} \qquad (43)$$

and

$$\alpha(\omega(\tau)) = j \text{ if and only if } \tau \cap (\{1,\ldots,m\}\times\{k\}) = \{(j,k)\} \text{ for all } \tau\varepsilon\lambda, \qquad (44)$$

using an arbitrary $\omega \varepsilon \mathcal{W}(\lambda,n)$ such that $\omega(\tau)=\omega'(\tau_{|k'})$ for every $\tau\varepsilon\lambda$ such that $\tau_{|k'}\neq\phi$.

<u>Proof</u>: See Appendix A.

The above theorem does not state how to start the recursion, i.e., a formula for the minimum k in K. For such a k, the left hand side of (40) is obtained by replacing $P_{\mathcal{H}}^{(k')}(\lambda_{|k'}|Z^{(k')})$ and $P_{N_T}^{(k')}(n|\lambda_{|k'},Z^{(k')})$ on the right hand side of (40) by 1 and Prob.$\{N_T=n\}$, resp., and replacing $P_X^{(k')}(dX'|\omega',n,\lambda_{|k'},Z^{(k')})$ in (42) by $Q_x^n(dX')$ with $\Delta t=t-t_0$.

In order to complete the recursion, we need the updating equations for $P_X^{(k)}$ and $P_{N_T}^{(k)}$. These equations are obtained by Bayesian expansions which are very similar to the one used to obtain (40) and are stated below without proof. Under Assumptions 1 - 8, for each k in K with an immediate predecessor k' and for each $\lambda\,\varepsilon\mathcal{H}(k)$, we have

$$P_X^{(k)}(dX|\omega,n,\lambda,Z^{(k)}) = (\mathcal{L}(y,m,n,\lambda,k))^{-1}P_M(y|\alpha,m,\delta,X,n,k)P_{N_{FA}}(m-n_D(\lambda|k)|\delta,X,n,k)P_D(\delta|X,n,k) \int_{\mathcal{X}_n} F_{\Delta t}^n(dX|X')P^{(k')}(dX'|\omega',n,\lambda_{|k'},Z^{(k')}) \qquad (45)$$

and

$$P_{N_T}^{(k)}(n|\lambda,Z^{(k)}) = \begin{cases} C_{N_T}^{(k)}(\lambda,Z^{(k)})^{-1} \dfrac{(n-\#(\lambda|_{k'}))!}{(n-\#(\lambda))!} \mathcal{L}(y,m,n,\lambda,k) P_{N_T}^{(k')}(n|\lambda|_{k'},Z^{(k')}) & \text{if } n \geq \#(\lambda) \\ 0 & \text{otherwise,} \end{cases} \tag{46}$$

where (ω,ω') and (α,δ) are chosen or determined in exactly the same way as in Theorem 1. The normalizing constant in (46) is given by

$$C_{N_T}^{(k)}(\lambda,Z^{(k)}) = \sum_{n=\#(\lambda)}^{\infty} \frac{(n-\#(\lambda|_{k'}))!}{(n-\#(\lambda))!} \mathcal{L}(y,m,n,\lambda,k) P_{N_T}^{(k')}(n|\lambda|_{k'},Z^{(k')}) \, . \tag{47}$$

Consequently, we have

$$P(Z^{(k)},\Lambda|_k|Z^{(k')},\Lambda|_{k'}) = \frac{N_{FA}(k)!}{N_M(k)!} C_{N_T}^{(k)}(\Lambda|_k,Z^{(k)}) . \tag{48}$$

When k is minimal, $P_X^{(k')}$ and $P_{N_T}^{(k')}$ in (45) and (46) should be replaced by Q_0^n and Prob.$\{N_T=n\}$ with $\Delta t=t-t_0$. Thus a fairly general multitarget tracking algorithm with the information state,

$$\left(\lambda,((P_X^{(k)}(\cdot|\omega,n,\lambda,Z^{(k)}),\ P_{N_T}^{(k)}(n|\lambda,Z^{(k)}))_{n=\#(\lambda)}^{\infty},P_{\mathcal{H}}^{(k)}(\lambda|Z^{(k)})\right)_{\lambda\epsilon\mathcal{H}} \ ,$$

has been completely described. With this algorithm we may, at least theoretically, handle complicated situations such as targets moving in a group. It is obvious, however, that implementation poses a serious problem. One of the difficulties is due to the high dimensionality of the information state of the multitarget tracker, which essentially covers all the $\mathcal{X}_n$'s, possibly from n=0 to infinity. The likelihood $\mathcal{L}$ defined by (42) must also be calculated for a great number of combinations of variables. Thus, even when we use extensive hypothesis reduction (management) techniques (which will be discussed in Part 2), further research may be necessary for implementing the general algorithm developed in this section.

On the other hand, if we introduce an appropriate set of independence assumptions, $P_X^{(k)}$ can be decomposed into a product of factors which can be shared among different hypotheses. More importantly, a finite set of distribution functions may cover all the $\mathcal{X}_n$'s. The likelihood $\mathcal{L}$ is also decomposed in a similar way. Roughly speaking, in such a case every evaluation can be done at the track level rather than at the hypothesis level. This will be clarified in the next section. Actually, almost all the existing multitarget literature assumes such a case. As discussed in the subsequent section, existing multitarget tracking algorithms can thus be viewed as being included in the general formula shown in this section as a subset.

V. Independent, Identically Distributed Targets

By i.i.d. (independent, identically distributed) target models, we actually mean a class of models for which several independence conditions are assumed. With such assumptions the general algorithm shown in the previous section can be greatly simplified since many terms can be reduced to the products of many factors which can be shared among other products.

We now assume that the target/sensor model satisfies the following additional set of assumptions:

Assumption A1:

For each positive integer n, we have $\mathcal{X}_n^C=\{0\}$ ($\mathcal{X}_n^C$ is ignored henceforth) and $\mathcal{X}_n^i=\mathcal{X}$, where $\mathcal{X}$ is a common hybrid space with a hybrid measure (Lebesgue-measure × counting measure), μ. By ignoring $\mathcal{X}_n^C$, the target system state space becomes

$$\mathcal{X}_n = (\mathcal{X})^n = \underbrace{\mathcal{X} \times \cdots\cdots \times \mathcal{X}}_{n} \quad . \tag{49}$$

Given $N_T=n$, $X=(x_i)_{i=1}^n$ is a system of time-homogeneous, independent and identically distributed Markov processes on $\mathcal{X}$ with the common a priori statistics defined by the initial distribution density,

$$\text{Prob.}\{x_i(t_0)\varepsilon dx\} = q_0(x)\mu(dx) \tag{50}$$

and the state transition probability density,

$$\text{Prob.}\{x_i(t+\Delta t)\varepsilon dx \mid x_i(t)=x'\} = f_{\Delta t}(x|x')\mu(dx) \quad . \tag{51}$$

In other words, we have

$$Q_0^n(\prod_{i=1}^{n} dx_i) = \prod_{i=1}^{n} q_0(x_i)\mu(dx_i) \tag{50'}$$

and

$$F_{\Delta t}^n(\prod_{i=1}^{n} dx_i \mid (x_i')_{i=1}^n) = \prod_{i=1}^{n} f_{\Delta t}(x_i|x_i')\mu(dx_i) \quad . \tag{51'}$$

<u>Assumption A2</u>:

The <u>a priori</u> distribution of N_T, the total number of targets, is Poisson with mean ν_0, i.e.,

$$\text{Prob.}\{N_T=n\} = \exp(-\nu_0)\frac{\nu_0^n}{n!} \quad . \tag{52}$$

<u>Assumption A3</u>:

The event pertaining to the detection of a target i depends only on its state x_i, i.e., for each k in K, each n, each $(x_i)_{i=1}^n \varepsilon\ \mathcal{X}^n$ and each $\delta\varepsilon\mathcal{D}(n)$, we have

$$P_D(\delta|(x_i)_{i=1}^n, n, k) = \prod_{i=1}^{n} p_D(x_i|k)^{\delta(i)} \ (1-p_D(x_i|k))^{(1\ -\ \delta(i))} \tag{53}$$

with a common detection probability function $p_D(\bullet|k):\mathcal{X} \to [0,1]$.

<u>Assumption A4</u>:

The number of false alarms for each data set k is independent of any target state or any other data set variable and has the distribution $p_{N_{FA}}(\cdot|k)$, i.e.,

$$p_{N_{FA}}(m|\delta,x,n,k) = p_{N_{FA}}(m|k) \qquad (54)$$

for all (δ,x,n,k). Given the number of false alarms in a data set $k=(t,s)$, the values of the false alarms are i.i.d. with the common probability density function $p_{FA}(\cdot|k)$ on $\mathcal{Y}_s$.

Assumption A5:

The measurement error in a measurement which originates from a target i in any data set $k=(t,s)$ depends only on the target state $x_i(t)$ and is modeled by a common transition probability density function $p_M(\cdot|\cdot,k):\mathcal{Y}_s\times\mathcal{X} \to [0,\infty)$ from $\mathcal{X}$ to $\mathcal{Y}_s$ for each k. Thus we have

$$p_M((y_j)_{j=1}^m|\alpha,m,\delta,(x_i)_{i=1}^n,n,k) = \left(\prod_{\substack{i=1\\ \delta(i)=1}}^{n} p_M(y_{\alpha(\{i\})}|x_i,k)\right)\cdot\left(\prod_{\substack{j=1\\ j\notin \mathrm{Image}(\alpha)}}^{m} p_{FA}(y_j|k)\right) \qquad (55)$$

for every $n\geq 0$, $(x_i)_{i=1}^n \in \mathcal{X}^n$, $\delta\in\mathcal{D}(n)$, $m\geq 0$, $\alpha\in\mathcal{A}^o(\{\{1\},\ldots,\{n\}\},\{1,\ldots,m\})$ and $(y_j)_{j=1}^m \in (\mathcal{Y}_s)^m$.

We should note that Assumptions A1 - A5 are assumptions which are "additional" to Assumptions 1 - 8. For example, equations (53) through (55) satisfy the requirement of Assumption 8. First let us discuss an important implication of the independence assumptions (49) to (55): For each k=(t,s) in K and each track $\tau \varepsilon \mathcal{T}(k)$, we define the <u>cumulative data set</u> $Z^{(k)}$ <u>restricted to track</u> τ by

$$Z^{(k)}_{|\tau} = \bigcup_{k' \preceq k} Y(\tau,k') \times \{k'\} \tag{56}$$

where $Y(\cdot,\cdot): \bigcup_{k\varepsilon K} \mathcal{T}^{(k)} \times K \rightarrow (\bigcup_{s\varepsilon S} \mathcal{Y}_s) \cup \{\theta\}$ is defined by

$$Y(\tau,k) = \begin{cases} \theta & \text{if } \tau \cap (J_M(k) \times \{k\}) = \phi \\ y_j(k) & \text{if } (j,k) \varepsilon \tau \end{cases} \tag{57}$$

for each $(\tau,k) \varepsilon \bigcup_{k\varepsilon K} \mathcal{T}^{(k)} \times K$, where $y_j(k)$ is the j-th measurement in data set k. The usage of θ is again symbolic, i.e., $(\theta,k) \varepsilon Z^{(k)}_{|\tau}$ means no measurement at k in track τ. On the other hand, $(y,t,s) \varepsilon Z^{(k)}_{|\tau}$ means that $y \varepsilon \mathcal{Y}_s$ is the measurement (value) which is hypothesized (by τ) to originate from a target creating track τ.

Then consider a Markov process x on $\mathcal{X}$, which is defined by q_0 and $f_{\Delta t}$, and an incomplete observation mechanism which creates a measurement $Y(\tau,k)$ if it succeeds in creating a measurement and provides nothing (represented by θ) if it fails, according to p_D and p_M described in (53) and (55), i.e., assume that

$$\text{Prob.}\{Y(\tau,k) \varepsilon dy | Y(\tau,k) \neq \theta, x(t)=x\} \; \text{Prob.}\{Y(\tau,k) \neq \theta | x(t)=x\} = p_M(y|x,k) p_D(x|k) \mu_s(dy) \tag{58}$$

and

$$\text{Prob.}\{Y(\tau,k)=\theta|x(t)=x\} = 1 - p_D(x|k) \quad . \tag{59}$$

Then the problem of calculating the state distribution of x at time t conditioned by $Z^{(k)}_{|\tau}$ (k=(t,s)), can be solved by applying standard filtering theory, i.e., by extrapolation using $f_{\Delta t}$ and updating using (58) or (59). Let us denote the solution to this "mini" or "single-target" problem by $p^{(k)}_{\tau}$, i.e., for every $\tau \varepsilon \bigcup_{k\varepsilon K} \mathcal{T}^{(k)}$, let

$$p^{(k)}_{\tau}(x)\mu(dx) = \text{Prob.}\{x(t)\varepsilon dx|Z^{(k)}_{|\tau}\} \quad . \tag{60}$$

The independence assumptions A1 to A5 then imply

$$\begin{aligned} p^{(k)}_{X}(\prod_{i=1}^{n} dx_i|\omega,n,\lambda,Z^{(k)}) &= \text{Prob.}\{X(t)\varepsilon \prod_{i=1}^{n} dx_i|\Omega_k=\omega,N_T=n,\Lambda_{|k}=\lambda,Z^{(k)}\} \\ &= \left(\prod_{\tau\varepsilon\lambda} p^{(k)}_{\tau}(x_{\omega(\tau)})\mu(dx_{\omega(\tau)})\right)\cdot\left(\prod_{i\not\varepsilon \text{Image}(\omega)} p^{(k)}_{\phi}(x_i)\mu(dx_i)\right) \end{aligned} \tag{61}$$

for each k in K, each $\lambda \varepsilon \mathcal{A}(k)$, each $n \geq \#(\lambda)$ and each $\omega \varepsilon \mathcal{W}(\lambda,n)$. Although (61) can be shown without difficulty, it is yet to be proven (actually, (61) is included in Theorem 2 below).

Let us introduce another useful notation: For each k=(t,s) in K, define the <u>track-measurement likelihood function</u> $L_k : (\mathcal{Y}_s \cup\{\theta\}) \times \mathcal{T}(k) \to [0,\infty)$ by

$$L_k(y,\tau) = \int_{\mathcal{X}} g_k(y|x)\tilde{p}^{(k)}_{\tau}(x)\mu(dx) \tag{62}$$

where $g_k(\cdot|\cdot):(\mathcal{Y}_s \cup \{\theta\})\times\mathcal{X} \to [0,\infty)$ is defined by

$$g_k(y|x) = \begin{cases} p_M(y|x,k)p_D(x|k) & \text{if } y\neq\theta \\ 1 - p_D(x|k) & \text{if } y=\theta \end{cases} \tag{63}$$

and, for every $\tau \in \mathcal{T}^{(k)}$,

$$\tilde{p}_\tau^{(k)}(x) = \begin{cases} \int_{\mathcal{X}} f_{t-t'}(x|x')p_\tau^{(k')}(x')\mu(dx') & \text{if k has an immediate predecessor } k'=(t',s') \\ \int_{\mathcal{X}} f_{t-t_0}(x|x')q_0(x')\mu(dx') & \text{otherwise.} \end{cases} \tag{64}$$

The main result of this section is shown below:

<u>Theorem 2</u>: Under Assumptions 1 to 8 and A1 to A5,

[1] For each k in K, we have

$$\text{Prob.}\{N_T=n|\Lambda_{|k},Z^{(k)}\} = \begin{cases} \exp(-\nu(k))\dfrac{\nu(k)^{(n - \#(\Lambda_{|k}))}}{(n - \#(\Lambda_{|k}))!} & \text{if } n\geq\#(\Lambda_{|k}) \\ 0 & \text{otherwise} \end{cases} \tag{65}$$

where $(\nu(k))_{k\in K}$ is given by

$$\nu(k) = \begin{cases} \nu(k')L_k(\theta,\phi) & \text{if k has an immediate predecessor } k' \\ \nu_0 L_k(\theta,\phi) & \text{if k is the minimum in K} \quad , \end{cases} \tag{66}$$

[2] for each k in K which has an immediate predecessor k' and each λ in $\mathcal{H}(k)$, when $Z(k)=((y_j)_{j=1}^m,m,k)$ is given, we have

$$p_{\mathscr{H}}^{(k)}(\lambda|Z^{(k)}) = p_{\mathscr{H}}^{(k')}(\lambda|_{k'}|Z^{(k')})\cdot\left[\prod_{\tau\varepsilon\lambda}(\nu(k'))^{\varepsilon(\tau)}L_k(Y(\tau,k),\tau)\right]$$

$$\cdot\left[\frac{((m-n_D(\lambda|k))!)p_{N_{FA}}(m-n_D(\lambda|k)|k)\prod_{j\varepsilon j_{FA}(\lambda,m|k)}p_{FA}(y_j|k)}{p_Z^{(k)}(Z^{(k)}|Z^{(k')})\cdot(m!)\cdot\exp(\nu(k')(1-L_k(\theta,\phi)))}\right] \tag{67}$$

where

$$j_{FA}(\lambda,m|k) = \left\{j\varepsilon\{1,\ldots,m\}\middle|\begin{matrix}\text{There is no }\tau\text{ in }\lambda\text{ such that}\\ \tau\cap(\{1,\ldots,m\}\times\{k\}) = \{(j,k)\}\end{matrix}\right\} \tag{68}$$

and

$$\varepsilon(\tau) = \begin{cases} 0 & \text{if } \tau|_{k'}\neq\phi \\ 1 & \text{if } \tau|_{k'}=\phi \end{cases} \tag{69}$$

whereas $n_D(\lambda|k)=m-\#(j_{FA}(\lambda,m|k))$ is as previously defined, and [3] for each k in K, we have (61) for each $\lambda\,\varepsilon\mathscr{H}(k)$, each $n\geq\#(\lambda)$ and each $\omega\,\varepsilon\mathscr{W}(\lambda,n)$, and moreover,

$$p_\tau^{(k)}(x) = L_k(Y(\tau,k),\tau)^{-1}g_k(Y(\tau,k)|x)\tilde{p}_\tau^{(k)}(x) \tag{70}$$

for each $\tau\,\varepsilon\mathscr{T}(k)$.

Proof: See Appendix B.

We should note that an empty track ϕ is always included in $\mathscr{T}(k)$ for any k in K according to our definition. Thus, $\tilde{p}_\phi^{(k)}(\cdot)$ is the a priori distribution density function (at k) which is common to all the undetected targets up to k (not including k), and $p_\phi^{(k)}(\cdot)$ is the a posteriori distribution density function (at k) of targets undetected up to and including k. The definition, (62) and (63), of the track-measurement likelihood functions L_k gives us the following verbal expression of (67): The posterior probability of any hypothesis is the product of

(1) a priori probability, $P_{\mathscr{H}}^{(k')}(\lambda_{|k'}|Z^{(k')})$, or the probability of the parent of λ,

(2) the likelihood of the set of measurements, $j_{FA}(\lambda)$, to be the false alarm set, $((m-n_D(\lambda|k))!)p_{N_{FA}}(m-n_D(\lambda|k)|k) \prod_{j \in j_{FA}(\lambda,m|k)} p_{FA}(y_j|k)$,

(3) the likelihood $L_k(Y(\tau,k),\tau)$ of measurement $Y(\tau,k)$ $(\neq\theta)$ originating from a previously detected target $(\tau_{|k'}\neq\phi)$

(4) the likelihood $L_k(Y(\tau,k),\tau)$ of a previously detected target $(\tau_{|k'}\neq\phi)$ being undetected $(Y(\tau,k)=\theta)$ and

(5) the likelihood $\nu(k')L_k(Y(\tau,k),\tau)$ of a measurement $Y(\tau,k)$ $(\neq\theta)$ originating from a newly detected target $(\tau_{|k'}=\phi)$

divided by the normalizing constant. Likewise, we may call $L_k(\theta,\phi)$ the likelihood of an undetected target remaining undetected.

As seen in (67) and (70), the evaluation of hypotheses can be done at the track level due to the independence assumptions. When k is the minimum in K, the left hand side of (67) can be calculated by replacing $P_Z^{(k)}(Z^{(k)}|Z^{(k')})$, $\nu(k')$, $\varepsilon(\tau)$ and $P_{\mathscr{H}}^{(k')}(\lambda_{|k'}|Z^{(k')})$ by $P_Z^{(k)}(Z^{(k)})$, ν_0, 1 and 1, resp. The initial condition for the filtering equation (70) is already included in the definition (64) of $\tilde{p}_\tau^{(k)}$. Thus we have given a complete description of the (so called) i.i.d. multitarget tracker, whose information state at k is

$$((\lambda, p_{\mathcal{H}}^{(k)}(\lambda|Z^{(k)}))_{\lambda \in \mathcal{H}(k)}, (\tau, p_{\tau}^{(k)})_{\tau \in \mathcal{T}(k)}, \nu(k)) \qquad (71)$$

that is to be propagated forward.

Although the algorithm shown in Theorem 2 is less general than that shown in Theorem 1, it covers nearly all the existing multitarget tracking algorithms as its subset,as shown in the next section. As is well known, the cardinality of $\mathcal{H}(k)$ and $\mathcal{T}(k)$ grows very rapidly. Hence any inplementation of the general algorithm described in this section requires further consideration. Such issues are discussed in Part 2.

VI. Relation to Existing Results

In this section, we use the i.i.d. model; i.e., we retain Assumptions A1 to A5 in addition to Assumptions 1 to 8. These assumptions are inherent in most of the multitarget tracking literature published thus far, although they are sometimes not stated explicitly.

Before discussing the relation of the general algorithm described by Theorem 2 to existing results, let us describe a batch-processing version of the same algcrithm. The theorem described below is easily obtained by applying (67) repeatedly. Hence the proof is omitted.

Theorem 3: Under Assumptions 1 to 8 and A1 to A5 with the notation, $n_{FA}(\lambda|k)=\#(j_{FA}(\lambda,N_M(k)|k)$, for any k in K and any λ in $\mathscr{H}(k)$, we have

$$P_{\mathscr{H}}^{(k)}(\lambda|Z^{(k)}) = C_B^{(k)}(Z^{(k)})^{-1}\cdot\ell_{FA}^{(k)}(\lambda)\cdot\prod_{\tau\in\lambda}\ell_\tau^{(k)} \qquad (72)$$

where

$$C_B^{(k)}(Z^{(k)}) = P_Z^{(k)}(Z^{(k)})\cdot(\prod_{k'\precsim k} N_M(k')!)\cdot\exp(\nu_0 - \nu(k)), \qquad (73)$$

$$\ell_{FA}^{(k)}(\lambda) = \prod_{k'\precsim k}(n_{FA}(\lambda|k')!)P_{N_{FA}}(n_{FA}(\lambda|k')|k')\prod_{j\in j_{FA}(\lambda|k')}p_{FA}(y_j(k')|k') \qquad (74)$$

and

$$\ell_\tau^{(k)} = \nu(\bar{k}(\tau))\prod_{\bar{k}(\tau)\precsim k'\precsim k} L_{k'}(Y(\tau,k'),\tau) = \nu_0\prod_{k'\precsim k} L_{k'}(Y(\tau,k'),\tau) \qquad (75)$$

for every $\tau\in\mathscr{T}(k)$ where $\bar{k}(\tau)$ is the minimum of set $\{k'\in K|k'\precsim k,\tau\cap(J_M(k')\times\{k'\})\neq\phi\}$, when $Z^{(k)} = \bigcup_{k' \preceq k} ((y_j(k'))_{j=1}^{N_M(k')}, N_M(k'), k')$.

It is possible to prove Theorem 3 without using Theorem 2 and to deduce (67) from (72). Theorem 3 states that a posteriori probability of each hypothesis λ at k is the product of the track likelihood $\ell_\tau^{(k)}$ of each track τ in λ and the false alarm likelihood divided by the normalizing constant, and provides us with a unified view of what we may call "track likelihood" approaches such as the algorithm described in [10] and [11]. The track likelihood updating equation, $\ell_\tau^{(k)} = \ell_{\tau|k'}^{(k')} L_k(Y(\tau,k),\tau)$, follows immediately from (75) with k' being the immediate predecessor of k. When each $\tilde{p}_\tau^{(k)}$ and p_M are gaussian, $L_k(y,\tau)$ $(y \neq \theta)$ is the exponent of the negative square innovations norm (times some constant), which is sometimes called "scores." (see [12].) As is well known, the squared innovations norm or its sum over a track may be considered a X^2 random variable.

Let us make one more assumption:

Assumption A4':

$p_{N_{FA}}(\cdot|k)$ is Poisson with mean $\nu_{FA}(k)$ for each k in K,

Then we can rewrite (72) as

$$P^{(k)}(\lambda|Z^{(k)}) = \tilde{C}_B^{(k)}(Z^{(k)})^{-1} \prod_{\tau\epsilon\lambda} \tilde{\ell}_\tau^{(k)} \qquad (72')$$

where

$$\tilde{C}_B^{(k)}(Z^{(k)}) = C_B^{(k)}(Z^{(k)}) \cdot \left[\prod_{k' \preceq k} \exp(-\nu_{FA}(k')) \prod_{j \epsilon N_M(k')} \nu_{FA}(k') p_{FA}(y_j(k')|k') \right]^{-1}$$

$$(73')$$

and

$$\tilde{\ell}_\tau^{(k)} = \nu_0 \prod_{k' \lesssim k} \tilde{L}_{k'}(Y(\tau,k),\tau) \tag{75'}$$

with

$$\tilde{L}_k(y,\tau) = \begin{cases} L_k(\theta,\tau) & \text{if } y=\theta \\ \dfrac{L_k(y,\tau)}{\nu_{FA}(k)p_{FA}(y|k)} & \text{if } y\neq\theta \end{cases} . \tag{76}$$

Take the logarithm of (72') and ignore the normalizing constant. Then we have a function h: $\mathscr{H}(k) \rightarrow (-\infty,\infty)$ defined by

$$h(\lambda) = \sum_{\tau\varepsilon\lambda} \log(\tilde{\ell}_\tau^{(k)}) = \sum_{\tau\,\varepsilon\,\mathscr{T}(k)\setminus\phi} \log(\tilde{\ell}_\tau^{(k)})\chi(\tau;\lambda) \tag{77}$$

for every λ in $\mathscr{H}(k)$ ($\chi(\cdot;A)$ is the indicator (characteristic function) of set A.). With (77), we can interpret Morefield's 0-1 integer programming algorithm described in [9]. Namely, the problem of obtaining the maximum a posteriori probability (MAP) hypothesis λ at k given $Z^{(k)}$ is equivalent to maximizing (77) with respect to λ in $\mathscr{H}(k)$. In (77), $(\chi(\tau;\lambda))_{\tau\,\varepsilon\,\mathscr{T}(k)\setminus\phi}$ is the 0-1 vector to which the 0-1 integer programming is applied. Then the constraint imposed by Assumptions 5 and 6 can be written in a 0-1 matrix-vector inequality as described in [9].

In most of the exicting multitarget tracking literature, in addition to Assumptions 1 to 8, A1 to A5 and A4', the following assumptions are made:

(1) $f_{\Delta t}$ is defined by a linear dynamical system driven by a white noise,

(2) $p_M(y|x,k)$ is defined by a linear-gaussian measurement equation,

$$y = H_k x + v_k$$

with an appropriate matrix H_K and an independent additive gaussian noise v_k, and (3) $p_{FA}(\cdot|k)$ is uniform over $\mathcal{Y}_s$.

In many cases, the initial distribution (modelled by $q_0(\cdot)$) of undetected targets as well as $\widetilde{p}_\phi(k)$ is relatively "uniform" or has a large variance when compared with the variance of the measurement noise v_k. Furthermore, if $p_D(\cdot|k)$ is constant over the field of view of each sensor, $p_\tau^{(k)}$ and $\tilde{p}_\tau^{(k)}$ with $\tau \neq \phi$ may be reasonably well approximated by gaussian densities with relatively small variances compared to the size of the field of view; hence, we have approximately

$$L_k(y,\tau) = \begin{cases} \pi_D \int_{\mathfrak{X}} p_M(y|x,k)\tilde{p}_\tau^{(k)}(x)\mu(dx) = \dfrac{\pi_D}{(2\pi)^{N/2}(\det(\Gamma))^{N/2}} \exp(-\tfrac{1}{2}||\tilde{e}_{k|k'}||_\Gamma^2 - 1) \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } y \neq \theta \\ 1 - \pi_D \qquad\qquad\qquad\qquad\qquad\quad \text{if } y = \theta \end{cases} \tag{78}$$

where $\tilde{e}_{k|k'} = y - H_k \hat{x}_{k|k'}$ is the innovations, $\Gamma = \mathrm{Var}(v_k) + H_k \Sigma_{k|k'} H_k^T$ is the innovations variance, $\hat{x}_{k|k'}$ and $\Sigma_{k|k'}$ are the mean and the variance of $\tilde{p}_\tau^{(k)}$, resp. ($||x||_A \underline{\underline{\Delta}} x^T A x$ and a^T is the transpose of vector or matrix a.), and π_D is the constant value of $p_D(\cdot|k)$. When updated by (70), an approximation similar to (78) leads us to the usual Kalman filter equation if $y \neq \theta$ and no updating ($p_\tau^{(k)} = \tilde{p}_\tau^{(k)}$) if $y = \theta$.

Therefore, with all the additional assumptions described above, it is easy to see that (67) becomes Reid's algorithm described in [4] with

$$\beta_{NT}(k) = \nu(k')L_k(y,\tau) \tag{79}$$

for τ such that $\tau|_{k'}=\phi$ where k' is the immediate predecessor of k. β_{NT} is called the "density of previously unknown targets that has been detected" in [4]. Moreover, the constant $p_D(\cdot|k)$ implies that $N_{DT} \triangleq \#(\{\tau \varepsilon \lambda | \tau|_{k'} \neq \phi \ , \ \tau \cap (J_M(k) \times \{k\}) \neq \phi\})$ has a binomial probability distribution given $N_{TGT} \triangleq \#(\lambda|_{k'})$, enabling us to use N_{DT} (among others) to expand (39), which is actually done in [4]. On the other hand, β_{NT} given by (79) should be a function of k and y. If it is fixed to a certain value, newly detected targets acquire increasingly (w.r.t. k) unjustifiably high possibilities. To prevent this from happening, Reid proposed to adjust β_{NT} as described in a paragraph in [4]:

> "....., a calculation of β_{NT}, the density of new (i.e., unknown) targets, is performed whenever a data set from a type 1 sensor is received. ...... β_{NT} depends upon the number of times the area has been observed by a type 1 sensor and possible flux of undetected targets into and out of the area."

Aside from this description, there is no further discussion of this calculation of β_{NT} in [4].

In contrast, according to our formulation, β_{NT} is analytically given by (79). In the original report [13] by Reid, a rather heuristic method for calculating β_{NT} is described, in which the sensor field of view is divided into many cells and the inflow/outflow from cell to cell of undetected targets is calculated. As seen in the previous section, however, the likelihood (79) of a measurement y originating from a newly detected target is calculated from $\nu(k')$ and $\tilde{p}_{\phi}^{(k)}$, both of which are calculated recursively for all k's. The exact calculation of $p_{\tau}^{(k)}$ or $\tilde{p}_{\tau}^{(k)}$, and accordingly (79), is, however, not generally possible due to the nonlinearity of $p_D(\cdot|k)$. Therefore effective approximation techniques must be exploited. With an appropriate approximation of $p_{\tau}^{(k)}$ and (79), we can properly consider the fact that a newly detected target would most likely appear on the edges of sensor fields of view and not in the middle. In many cases, when only a small number of measurement indices are in a track τ, p_{τ} or $\tilde{p}_{\tau}$ may not be well approximated by gaussian densities. For example, consider a case where targets moves in a 1-dimensional space, the a priori target velocity information contained in $q_0(\cdot)$ is represented by a uniform distribution on a possible velocity range and there is no velocity measurement (position-only measurement). In such a case, gaussian approximations of p_{τ} or $\tilde{p}_{\tau}$ are very poor. Appropriate approximation methods are, therefore, called for in order to calculate p_{τ} and $\tilde{p}_{\tau}$, and accordingly, the likelihood function L_k. As mensioned in [4], such approximations coupled with hypothesis management techniques (described in Part 2) can be viewed as so called "track initiation processes."

On the other hand, when a separate track initiation mechanism is assumed, such as in [14], one of the most difficult parts of the multitarget tracking problem is removed automatically. Any tracking algorithm with a separate track initiator can be incorporated into our framework as follows: First extend the set S of sensors to $\bar{S}=\{s_0\}\cup S$ where s_0 is a track initiator as a "super" sensor which creates a probability-one hypothesis, $\lambda_0=\{\bar{\tau}_1,\ldots,\bar{\tau}_{\bar{N}}\}$ with $\bar{\tau}_i=(i,t_0,s_0)$ being the i-th a priori track. Then, if we replace K by $\bar{K}=K\cup(t_0,s_0)$, Theorem 2 or 3 provides a general Bayesian formula for cases in which a separate track initiator is employed. Those cases in which a track initiator provides new tracks in the middle as well as the beginning of the tracking may be similarly incorporated. Although the use of a tracker and a track initiator in parallel can be handled in our general framework, such a use may be seen as a departure from a purely Bayesian approach. A track initiator does have memory. Unless the track initiator does not share the data sets with a tracker, the correlation between the data sets and the outputs of the track initiator cannot be ignored. Therefore, we must either divide the data sets, one for the track initiator and the other for the tracker, or use more or less heuristic methods to discount such an effect as double counting or too much reliance. For this reason, we may say multitarget trackers used with separate track initiators are either restrictive or "sub-optimal."

As discussed in greater detail in Part 2, since $\mathscr{H}(k)$ is the collection of mutually distinct and collectively exhaustive hypotheses, aggregation or combining of hypotheses, such as $\left\{\Lambda_{|k}=\lambda_1 \text{ or } \Lambda_{|k}=\lambda_2\right\}$, is compatible with our formulation and is a great help from the view point of implementation. However, in order to perform such operations properly, we must know the correspondence among tracks in hypotheses to be combined. When we assume a separate track initiator and there is no newly detected target, such correspondence is obvious and it is possible to combine all the hypotheses so that there is always only one (and hence probability-one) hypothesis to be propagated forward. The JPDA (Joint Probabilistic Data Association) method described in [14] thoroughly exploits such condition.

One of the points which we have stressed in the previous two sections is that multitarget tracking requires a hierarchical algorith; the evaluation of hypotheses is at the top and (generally nonlinear) filtering at the bottom. Thus, the construction of multitarget trackers in many different situations creates a wide range of nonlinear filtering problems. The use of hybrid-state

Markovian models enables us to treat a wide range of complicated situations, at least in principle. Fairly complicated dynamics are occasionally used in multitarget tracking literature, e.g., [15] with birth-death processes. To the best of our knowledge, however, there is still no satisfactory nonlinear filtering for maneuvering targets. When there is no discrete-part dynamics (target classification problems, etc.), the required filtering is substantially simple. Particularly, if, in addition, the continuous-part dynamics is linear-Gaussian, the sum-of-Gaussian filtering described in [17] may be the most appropriate. On the other hand, even when there is no discrete-part state, and all track statistics are Gaussian, because of the huge computational requirement generally associated with any multigarget tracker, efforts to develop filtering techniques by which each track-measurement likelihood can be quickly calculated are always worthwhile. One of such efforts is described in [16].

VII. Conclusion

According to our viewpoint, targets and sensors in a general multitarget tracking environment are properly modeled only when targets are modeled as a random-set process and each sensor is regarded as a mechanism which maps this set to other random sets, i.e., measurement data sets. A very general target/sensor model has been defined and a general recursive multitarget tracking algorithm has been derived based upon this viewpoint of ours and Bayes' rule. Then a special case with a so-called i.i.d. (target) model, has been examined in more detail, and a general multitarget tracking algorithm both in recursive and batch-processing forms have been derived. Besides the generality of (individual) target dynamics and sensor models, two previously ignored but realistically very important factors have been pointed out: (1) state-dependent probability of detection and (2) precise definition of likelihood of a measurement originating from a newly detected target. Our general i.i.d. tracking algorithm has been been compared with existing algorithms which share the common concepts of tracks and hypotheses. We have succeeded in providing a unified view of existing algorithms by showing that the general algorithm is in fact a generalization of many well-known algorithms. We have also shown that the general multitracking algorithm is hierarchical in nature and always contain a nonlinear (or linear) filtering algorithm as a sub-algorithm.

This paper, Part 1, covers most of our theoretical developments on multitarget tracking. Part 2 of this paper will consider hypothesis management techniques and other implementatin issues. The term "hypothesis management" is borrowed from the artificial intelligence (AI) terminology. In our context it means a set of procedures which keep the number of hypotheses, and hence, a multitarget tracking algorithm under control. Due to the rapid growth of the number of hypotheses, no multitarget tracking algorithm is implementable without appropriate hypothesis management procedures. At Advanced Information & Decision Systems (AI&DS), we have developed a system called GTC (Generalized Tracker/Classifier) which implements all the problem-independent parts of the general (i.i.d.) tracking algorithm. In Part 2 we will present some numerical results to illustrate the use of this system.

REFERENCE

[1] Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment," IEEE Trans. Automat. Contr., vol. AC-23, pp. 618-626, Aug. 1978.

[2] H.L. Wiener, W.W. Willman, I.R. Goodman and J.H. Kullback, "Naval Ocean-Surveillance Correlation Handbook, 1978," NRL Report 8340, Naval Research Lab., Washington, D.C., 1979.

[3] I.R. Goodman, H.L. Wiener and W.W. Willman, "Naval Ocean Surveillance Correlation Handbook, 1979," NRL Report 8402, Naval Research Lab., Washington, D.C., 1980.

[4] D.B. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. Automat. Contr., vol. AC-24, No. 6, pp. 843-854, Dec., 1979.

[5] K.R. Pattipati, N.R. Sandell, Jr. and L.C. Kramer, "A Unified View of Multi-Object Tracking," Proc. 4th MIT/ONR Workshop on Distributed Information and Decesion Systems Motivated by Command-Control-Communications Problems, vol. 1, pp. 115-135, San Diego, Ca., June 1981.

[6] G. Matheron, Random Sets and Integral Geometry, John Wiley & Sons, Inc., 1975.

[7] R.P. Hughes and R.R. Tenney, "A Multiobject Tracking Algorithm for a Distributed Sensor Network," Proc. 3rd MIT/ONR Workshop on Distributed Information and Decesion Systems, Sept. 1980.

[8] K.-C. Chang and Y. Bar-Shalom, "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements," to appear in IEEE Tranc. Automat. Contr.

[9] C.L. Morefield, "Application of 0-1 Integer Programming to Multitarget Tracking Problems," IEEE Trans. Automat. Contr., vol. AC-22, no. 3, pp. 302-312, June 1977.

[10] K.M. Keverian, "Multi-Object Tracking by Adaptive Hypothesis Testing," S.B. Thesis, M.I.T., Cambridge, MA., May 1979.

[11] K.M. Keverian and N.R. Sandell, Jr., "Multi-Object Tracking by Adaptive Hypothesis Testing," Report LIDS-R-959, MIT, Dec. 1979.

[12] D.L. Alspach and R.N. Lobbia, "A Score for Correct Data Association in Multi-Target Tracking," Proc. 18th IEEE Conf. Decision and Control, Fort Lauderdale, FL., Dec. 1979.

[13] D.B. Reid, "A Multiple Hypothesis Filter for Tracking Multiple Targets in a Cluttered Environment," Report LMSC-D560254, Lockheed Missiles & Space Co., Inc., Sunnyvale, CA., Sept. 1977.

[14] T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Joint Probabilistic Data Association for Multiple Targets in Clutter," Proc. 19th IEEE Conf. Decision and Control, Albuquerque, NM., Dec. 1980.

[15] I.R. Goodman, "A General Model for the Multiple Target Correlation and Tracking Problem," Proc. 18th IEEE Conf. Decision and Control, Fort Lauderdale, FL., Dec. 1979.

[16] H.W. Sorenson, "An Efficient Track Association Algorithm for the Multitarget Tracking Problem," Information Sciences, 21, 241-260(1980).

[17] D.L. Alspach, "A Gaussian Sum Approach to the Multi-Target Identification-Tracking Problem," Automatica, vol. 11. pp. 285-296, 1975.

Appendix A: (Proof of Theorem 1)

As mentioned in the early part of Section IV, the most crucial part of the proof of the theorem is how to calculate or expand $P(Z^{(k)},\Lambda_{|k}|Z^{(k')},\Lambda_{|k'})$, which is outlined in (30) and (39). The precise meaning of (39) is

$$\text{Prob.}\{y(k)\varepsilon Y, N_M(k)=m, \Lambda_{|k}=\lambda|\Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\} = \sum_{n=\#(\lambda)}^{\infty} \text{Prob.}\{N_T=n|\Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\}$$

$$\sum_{\omega'\,\varepsilon\mathcal{W}(\lambda_{|k'},n)} \text{Prob.}\{\Omega_{k'}=\omega'|N_T=n, \Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\}$$

$$\sum_{\omega\,\varepsilon\mathcal{W}(\lambda,n)} \text{Prob.}\{y(k)\varepsilon Y, N_M(k)=m, \Lambda_{|k}=\lambda, \Omega_k=\omega|\Omega_{k'}=\omega', N_T=n, \Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\} \tag{A1}$$

for every $\lambda\,\varepsilon\mathcal{A}(k)$, every $m\geq 0$ and every measurable set Y in $(\mathcal{Y}_S)^m$. The first and second terms are already given by (37) (part of recursive assumptions) and by (33). The third term in (A1) is further expanded as

$$\text{Prob.}\{y(k)\varepsilon Y, N_M(k)=m, \Lambda_{|k}=\lambda, \Omega_k=\omega|\Omega_{k'}=\omega', N_T=n, \Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\} =$$

$$\int_{\mathcal{X}_n} \text{Prob.}\{y(k)\varepsilon Y, N_M(k)=m, \Lambda_{|k}=\lambda, \Omega_k=\omega|X(t)=X, \Omega_{k'}=\omega', N_T=n, \Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\}$$
$$\text{Prob.}\{X(t)\varepsilon dX|\Omega_{k'}=\omega', N_T=n, \Lambda_{|k'}=\lambda_{|k'}, Z^{(k')}\}\ . \tag{A2}$$

The conditional probability measure on $\mathcal{X}_n$ in (A2) is given by (38) (part of the recursive assumptions), i.e., we have

$$\text{Prob.}\{X(t)\varepsilon dX|\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\} = \int_{\mathfrak{X}_n} F^n_{\Delta t}(dX|X')P_X^{(k')}(dX'|\omega',n,\lambda_{|k'},Z^{(k')}) \tag{A3}$$

with $\Delta t=t-t'$.

On the other hand, the integrand in (A2) can be further expanded as

$$\begin{aligned}
&\text{Prob.}\{y(k)\varepsilon Y,N_M(k)=m,\Lambda_{|k}=\lambda,\Omega_k=\omega|X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\\
&= \sum_{\delta\,\varepsilon\,\mathcal{D}} \text{Prob.}\{F_D(k)=\delta|X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\\
&\quad \text{Prob.}\{N_M(k)=m|F_D(k)=\delta,X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\\
&\sum_{\alpha\varepsilon\mathcal{A}^o(\{\{i\}|\delta(i)=1\},\{1,..,m\})} \text{Prob.}\{A(k)=\alpha|N_M(k)=m,F_D(k)=\delta,X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\\
&\quad \text{Prob.}\{\Lambda_{|k}=\lambda|A(k)=\alpha,N_M(k)=m,F_D(k)=\delta,X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\\
&\quad \text{Prob.}\{\Omega_k=\omega|A(k)=\alpha,N_M(k)=m,F_D(k)=\delta,X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\\
&\text{Prob.}\{y(k)\varepsilon Y|\Omega_k=\omega,A(k)=\alpha,N_M(k)=m,F_D(k)=\delta,X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}.
\end{aligned} \tag{A4}$$

Under Assumptions 1-8, the first, the second, the third and the sixth terms are given by (15), (16), (19) and (20). The fourth and the fifth terms merely check the consistency among $(\lambda,\omega,\omega',\alpha)$. The fourth term, $\text{Prob.}\{\Lambda_{|k}|\ldots\}$, is 1 if

$$\lambda = \left(\bigcup_{\tau'\varepsilon\lambda_{|k'}} \{\tau'\bigcup\{(\alpha(\{\omega'(\tau')\}),k)\}\}\right) \bigcup \left(\bigcup_{i\notin \text{Image}(\omega')} \{(\alpha(\{\omega'(\tau')\}),k)\}\right) \tag{A5}$$

and 0 otherwise. Likewise the fifth term, Prob.$\{\Omega_k=\omega|..\}$, is 1 if

$$\omega(\tau) = \begin{cases} \omega'(\tau_{|k'}) & \text{if } \tau_{|k'}\neq\phi \\ \text{a unique } i \text{ such that} \\ \tau = \{(\alpha(\{i\}),k)\} & \text{otherwise} \end{cases} \tag{A6}$$

and is 0 otherwise.

When a subset $\{\tau\varepsilon\lambda|\tau_{|k'}=\phi\}$ of new tracks in λ is not empty and

$$n - \#\{\tau\varepsilon\lambda|\tau_{|k'}\neq\phi\} > \#\{\tau\varepsilon\lambda|\tau_{|k'}=\phi\} ,$$

there are more than one (α,δ) which satisfies $\alpha\varepsilon\mathscr{A}^O(\{\{i\}|\delta(i)=1\},\{1,..,m\})$ and (A5). On the other hand, for given λ and ω, there exists one and only one (α,δ) such that $\alpha\varepsilon\mathscr{A}^O(\{\{i\}|\delta(i)=1\},\{1,..,m\})$ and (A6) hold.

Therefore, for any $m\geq0$, for any measurable set Y in $(\mathcal{Y}_s)^m$, any $\lambda\,\varepsilon\mathscr{H}(k)$, any $n\geq\#(\lambda)$, any $x\,\varepsilon\mathcal{X}_n$ and any $\omega'\,\varepsilon\mathcal{W}(\lambda_{|k'},n)$, if $\omega\,\varepsilon\mathcal{W}(\lambda,n)$ satisfies

$$\omega(\tau) = \omega'(\tau_{|k'}) \quad \text{for all } \tau\varepsilon\lambda \text{ such that } \tau_{|k'}\neq\phi \quad , \tag{A7}$$

we have

$$\text{Prob.}\{y(k)\varepsilon Y,N_M(k)=m,\Lambda_{|k}=\lambda,\Omega_k=\omega|\,X(t)=X,\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}$$

$$= P_D(\delta|X,n,k)\,P_{N_{FA}}(m-n_D(\lambda|k)|\delta,X,n,k)\frac{(m-n_D(\lambda|k))!}{m!}\int_Y P_M(y|\alpha,m,\delta,X,n,k)\,\mu_s^m(dy) \tag{A8}$$

where $n_D()$, δ and α are defined by (41), (43) and (44), resp. Otherwise the left hand side of (A8) is zero. Substitute (A8) into (A2) and perform integration using (A3). Then the integral (A2) has the same value for every

(ω,ω') in $\mathscr{W}(\lambda,n)\times\mathscr{W}(\lambda_{|k'},n)$ as long as (A7) is satisfied, due to Assumptions 2 and 8.

Suppose $\lambda\,\varepsilon\,\mathscr{H}(k)$, $n\geq\#(\lambda)$ and $\omega'\,\varepsilon\,\mathscr{W}(\lambda_{|k'},n)$. Then there exist $\binom{n\;-\;\#(\lambda_{|k'})}{\#(\lambda)-\#(\lambda_{|k'})}$ combinations to choose sets of newly detected target indices for $\{\tau\varepsilon\lambda|\tau_{|k'}=\phi\}$ and, for each of such combinations, there exists $(\#(\lambda)-\#(\lambda_{|k'}))!$ isomorphisms from a chosen target index set to $\{\tau\varepsilon\lambda|\tau_{|k'}=\phi\}$. Therefore,

$$\binom{n\;-\;\#(\lambda_{|k'})}{\#(\lambda)-\#(\lambda_{|k'})}(\#(\lambda)\;-\;\#(\lambda_{|k'}))!\;=\;\frac{(n\;-\;\#(\lambda_{|k'}))!}{(n\;-\;\#(\lambda))!} \tag{A9}$$

is the number of ω's in $\mathscr{W}(\lambda,n)$ which satisfy (A7) for fixed $\lambda\varepsilon\mathscr{H}(k)$, $n\geq\#(\lambda)$, and $\omega'\varepsilon\,\mathscr{W}(\lambda_{|k'},n)$. On the other hand, when we replace k by k' in (33), we have

$$\text{Prob.}\{\Omega_{k'}=\omega'|N_T=n,\Lambda_{|k'}=\lambda_{|k'}\}=(\#\,\mathscr{W}(\lambda_{|k'},n))^{-1}.$$

Consequently (A1) is reduced to

$$\begin{aligned}&\text{Prob.}\{y(k)\varepsilon Y,N_M(k)=m,\Lambda_{|k}=\lambda|\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}=\\&\quad\sum_{n=\#(\lambda)}^{\infty}\text{Prob.}\{N_T=n|\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}.\frac{(n\;-\;\#(\lambda_{|k'}))!}{(n\;-\;\#(\lambda))!}\\&\qquad\qquad\text{Prob.}\{y(k)\varepsilon Y,N_M(k)=m,\Lambda_{|k}=\lambda,\Omega_k=\omega|\Omega_{k'}=\omega',N_T=n,\Lambda_{|k'}=\lambda_{|k'},Z^{(k')}\}\end{aligned} \tag{A10}$$

for any $\omega'\,\varepsilon\mathscr{W}(\lambda_{|k'},n)$ and any $\omega\,\varepsilon\mathscr{W}(\lambda,n)$ which satisfies (A7).

Substitute (A8) with $n_D()$, δ and α defined by (40), (43) and (44), resp., and take density on $(\mathcal{Y}_s)^m$ with respect to μ_s^m. Attach the prior

$$P_{\mathfrak{A}}^{(k')}(\lambda_{|k'}|Z^{(k')}) = \text{Prob.}\{\Lambda_{|k'}=\lambda_{|k'}|Z^{(k')}\}$$

to it and divide it by the normalizing constant $P_Z^{(k)}(Z^{(k)}|Z^{(k')})$. Then, we have (40). Q.E.D.

Appendix B: (Proof of Theorem 2)

We will prove the three parts [1] - [3], more or less simultaneously. Part [1] and part [3] will be proved by recursion. First, let us assume that (61) and (65) hold with k replaced by k'. By (61') or (65'), we mean (61) or (65) with k' instead of k. For any $\lambda \varepsilon \mathcal{A}(k)$, any $n \geq \#(\lambda)$, any $m \geq n_D(\lambda|k)$ and any $(y_j)_{j=1}^m \varepsilon (\mathcal{Y}_s)^m$, it follows from (51'), (53) - (55), (57), (61') and (62) - (64) that

$$
\begin{aligned}
&\mathcal{L}((y_j)_{j=1}^m, m, n, \lambda, k) \\
&= p_{N_{FA}}(m-n_D(\lambda|k)|k) \int_{\mathfrak{X}^n} \Big(\prod_{\substack{i=1 \\ \delta(i)=1}}^{n} p_M(y_{\alpha(\{i\})}|x_i, k) \cdot \prod_{\substack{j=1 \\ j \notin \mathrm{Image}(\alpha)}}^{m} p_{FA}(y_j|k) \Big) \\
&\qquad \cdot \Big(\prod_{i=1}^{n} p_D(x_i|k)^{\delta(i)} (1-p_D(x_i|k))^{(1-\delta(i))} \Big) \\
&\qquad \cdot \Big(\prod_{\substack{\tau \varepsilon \lambda \\ \tau|_{k'} \neq \phi}} \tilde{p}_\tau^{(k)}(x_{\omega'(\tau)}) \mu(dx_{\omega'(\tau)}) \cdot \prod_{i \notin \mathrm{Image}(\omega')} \tilde{p}_\phi^{(k)}(x_i) \mu(x_i) \Big) \\
&= p_{N_{FA}}(m-n_D(\lambda|k)|k) \cdot \prod_{\substack{j=1 \\ j \notin \mathrm{Image}(\alpha)}}^{m} p_{FA}(y_j|k) \\
&\qquad \cdot \Big[\prod_{\tau \varepsilon \lambda \cap \mathcal{J}_k^m} \int_{\mathfrak{X}} p_M(y_{\alpha(\{\omega(\tau)\})}|x, k) p_D(x|k) \tilde{p}_\tau^{(k)}(x) \mu(dx) \Big] \\
&\qquad \cdot \Big[\prod_{\tau \varepsilon \lambda \setminus \mathcal{J}_k^m} \int_{\mathfrak{X}} (1 - p_D(x|k)) \tilde{p}_\tau^{(k)}(x) \mu(dx) \Big] \\
&\qquad \cdot \Big[\int_{\mathfrak{X}} (1 - p_D(x|k)) \tilde{p}_\phi^{(k)}(x) \mu(dx) \Big]^{(n - \#(\lambda))} \\
&= p_{N_{FA}}(m-n_D(\lambda|k)|k) \cdot \prod_{j \varepsilon j_{FA}(\lambda, m|k)} p_{FA}(y_j|k) \cdot \prod_{\tau \varepsilon \lambda} L_k(Y(\tau, k), \tau) \cdot L_k(\theta, \phi)^{(n-\#(\lambda))} \qquad \text{(B1)}
\end{aligned}
$$

where $J_k^m=\{1,\dots,m\}\times\{k\}$, (ω,ω') in $\mathcal{W}(\lambda,n)\times\mathcal{W}(\lambda_{|k'},n)$ is an arbitrary pair satisfyin (A7), and $n_D(\lambda|k)$, $j_{FA}(\lambda,m|k)$, δ and α are defined by (41), (68), (43) and (44), resp.

According to Theorem 1 (Equation (40)), it follows from (B1) and (65') that

$$P_{\mathcal{H}}^{(k)}(\lambda|Z^{(k)}) = \frac{P_{\mathcal{H}}^{(k')}(\lambda_{|k'},|Z^{(k')})}{P_Z^{(k)}(Z^{(k)}|Z^{(k')})}\cdot\frac{(m-n_D(\lambda|k))!}{m!}\cdot p_{N_{FA}}(m-n_D(\lambda|k)|k)$$

$$\cdot\left[\prod_{j\in j_{FA}(\lambda,m|k)} p_{FA}(y_j|k)\right]\cdot\left[\prod_{\tau\in\lambda} L_k(Y(\tau,k),\tau)\right]$$

$$\cdot\exp(-\nu(k'))\cdot\exp(\nu(k')L_k(\theta,\phi))\cdot(\nu(k'))^{(\#(\lambda)-\#(\lambda_{|k'}))} \qquad \text{(B2)}$$

(67) follows immediately from (B2), and hence, part [2] has been proved.

It follows from (B1), (46) and (65') that, for any $n\geq\#(\lambda)$,

$$P_{N_T}^{(k)}(n|\lambda,Z^{(k)}) = \tilde{C}_{N_T}^{(k)}(\lambda,Z^{(k)})^{-1}\cdot\exp(-\nu(k'))\cdot\nu(k')^{(\#(\nu)-\#(\nu_{|k'}))}$$

$$\cdot\frac{(\nu(k')L_k(\theta,\phi))^{(n-\#(\lambda))}}{(n-\#(\lambda))!} \qquad \text{(B3)}$$

where $\tilde{C}_{N_T}^{(k)}(\lambda,Z^{(k)})$ is the normalizing constant. (B3) proves part [1] since (B3) holds true even when k is the minimal in K by letting $\lambda_{|k'}=\phi$ and $\nu(k')=\nu_0$.

(70) is an obvious consequence of the definitions, (60), (62), (63) and (64), of $p_\tau^{(k)}$, L_k, g_k and $\tilde{p}_\tau^{(k)}$, resp. (61) follows from (45), the assumptions, (51'), (53), (55) and (61'), and the definition of $p_\tau^{(k)}$. Q.E.D.

where $J_k^m=\{1,\ldots,m\}\times\{k\}$, (ω,ω') in $\mathcal{W}(\lambda,n)\times\mathcal{W}(\lambda_{|k'},n)$ is an arbitrary pair satisfying (A7), and $n_D(\lambda|k)$, $j_{FA}(\lambda,m|k)$, δ and α are defined by (41), (68), (43) and (44), resp.

According to Theorem 1 (Equation (40)), it follows from (B1) and (65') that

$$P_{\mathcal{H}}^{(k)}(\lambda|Z^{(k)}) = \frac{P_{\mathcal{H}}^{(k')}(\lambda_{|k'}|Z^{(k')})}{P_Z^{(k)}(Z^{(k)}|Z^{(k')})}\cdot\frac{(m-n_D(\lambda|k))!}{m!}\cdot p_{N_{FA}}(m-n_D(\lambda|k)|k)$$
$$\cdot\left[\prod_{j\varepsilon j_{FA}(\lambda,m|k)} p_{FA}(y_j|k)\right]\cdot\left[\prod_{\tau\varepsilon\lambda} L_k(Y(\tau,k),\tau)\right]$$
$$\cdot\ \exp(-\nu(k'))\cdot\exp(\nu(k')L_k(\theta,\phi))\cdot(\nu(k'))^{(\#(\lambda)\ -\ \#(\lambda_{|k'}))} \qquad \text{(B2)}$$

(67) follows immediately from (B2), and hence, part [2] has been proved.

It follows from (B1), (46) and (65') that, for any $n\geq\#(\lambda)$,

$$P_{N_T}^{(k)}(n|\lambda,Z^{(k)}) = \tilde{C}_{N_T}^{(k)}(\lambda,Z^{(k)})^{-1}\cdot\exp(-\nu(k'))\cdot\nu(k')^{(\#(\nu)-\#(\nu_{|k'}))}$$
$$\cdot\ \frac{(\nu(k')L_k(\theta,\phi))^{(n\ -\ \#(\lambda))}}{(n\ -\ \#(\lambda))!} \qquad \text{(B3)}$$

where $\tilde{C}_{N_T}^{(k)}(\lambda,Z^{(k)})$ is the normalizing constant. (B3) proves part [1] since (B3) holds true even when k is the minimal in K by letting $\lambda_{|k'}=\phi$ and $\nu(k')=\nu_0$.

(70) is an obvious consequence of the definitions, (60), (62), (63) and (64), of $p_\tau^{(k)}$, L_k, g_k and $\tilde{p}_\tau^{(k)}$, resp. (61) follows from (45), the assumptions, (51'), (53), (55) and (61'), and the definition of $p_\tau^{(k)}$. Q.E.D.

where $J_k^m=\{1,\ldots,m\}\times\{k\}$, (ω,ω') in $\mathcal{W}(\lambda,n)\times\mathcal{W}(\lambda_{|k'},n)$ is an arbitrary pair satisfying (A7), and $n_D(\lambda|k)$, $j_{FA}(\lambda,m|k)$, δ and α are defined by (41), (68), (43) and (44), resp.

According to Theorem 1 (Equation (40)), it follows from (B1) and (65') that

$$P_{\mathcal{H}}^{(k)}(\lambda|Z^{(k)}) = \frac{P_{\mathcal{H}}^{(k')}(\lambda_{|k'}|Z^{(k')})}{P_Z^{(k)}(Z^{(k)}|Z^{(k')})}\cdot\frac{(m - n_D(\lambda|k))!}{m!}\cdot p_{N_{FA}}(m-n_D(\lambda|k)|k)$$

$$\cdot\left[\prod_{j\in j_{FA}(\lambda,m|k)} p_{FA}(y_j|k)\right]\cdot\left[\prod_{\tau\in\lambda} L_k(Y(\tau,k),\tau)\right]$$

$$\cdot\ \exp(-\nu(k'))\cdot\exp(\nu(k')L_k(\theta,\phi))\cdot(\nu(k'))^{(\#(\lambda)\ -\ \#(\lambda_{|k'}))} \qquad \text{(B2)}$$

(67) follows immediately from (B2), and hence, part [2] has been proved.

It follows from (B1), (46) and (65') that, for any $n\geq\#(\lambda)$,

$$P_{N_T}^{(k)}(n|\lambda,Z^{(k)}) = \tilde{C}_{N_T}^{(k)}(\lambda,Z^{(k)})^{-1}\cdot\exp(-\nu(k'))\cdot\nu(k')^{(\#(\nu)-\#(\nu_{|k'}))}$$

$$\cdot\ \frac{(\nu(k')L_k(\theta,\phi))^{(n\ -\ \#(\lambda))}}{(n\ -\ \#(\lambda))!} \qquad \text{(B3)}$$

where $\tilde{C}_{N_T}^{(k)}(\lambda,Z^{(k)})$ is the normalizing constant. (B3) proves part [1] since (B3) holds true even when k is the minimal in K by letting $\lambda_{|k'}=\phi$ and $\nu(k')=\nu_0$.

(70) is an obvious consequence of the definitions, (60), (62), (63) and (64), of $p_\tau^{(k)}$, L_k, g_k and $\tilde{p}_\tau^{(k)}$, resp. (61) follows from (45), the assumptions, (51'), (53), (55) and (61'), and the definition of $p_\tau^{(k)}$. Q.E.D.

APPENDIX B

DISTRIBUTED ESTIMATION IN NETWORKS

C.Y. Chong, E. Tse and S. Mori

ABSTRACT

In this paper, we consider the distributed estimation problem by a set of agents connected by an arbitrary communication network. The agents communicate conditional probabilities of the random state over the network. From these conditional probabilities, each agent then tries to re-construct the conditional probability given all the measurements if these were communicated instead of the probabilities. It is discovered that in general the agents have to remember some of the past conditional probabilities and may even have to request additional information. A method for generating the fusion algorithm for each agent based on the network structure is presented and applied to some examples. The results are applicable to both dynamic and static states.

1. INTRODUCTION

The traditional approach to estimation has been centralized. Even though the measurements are generated by a large number of sensors, it is usually assumed that they are sent to a central site where processing is carried out by one agent (computer). In this context centralized estimation theory is well developed and has found applications in many real world problems.

In recent years, there has been growing interest in distributed estimation problems. In such problems (Figure 1), the sensor measurements are not all transmitted to a central processor. Instead, a set of local processors, which we call estimation agents, are present. The agents are connected by a communication network. Each agent collects the measurements from a subset of the sensors, performs some local processing, and communicates the results with other agents.

The advantages of such a distributed estimation system are many. It is more reliable (or less vulnerable) since there is not a single central site which is responsible for the proper functioning of the system. Communication is cheaper since only the results of processing, and not the raw data, are communicated. Furthermore, each distributed agent has the use of the processed data locally and does not have to wait for communication from the central processor. From a technological point of view, such distributed systems are made possible by the availability of

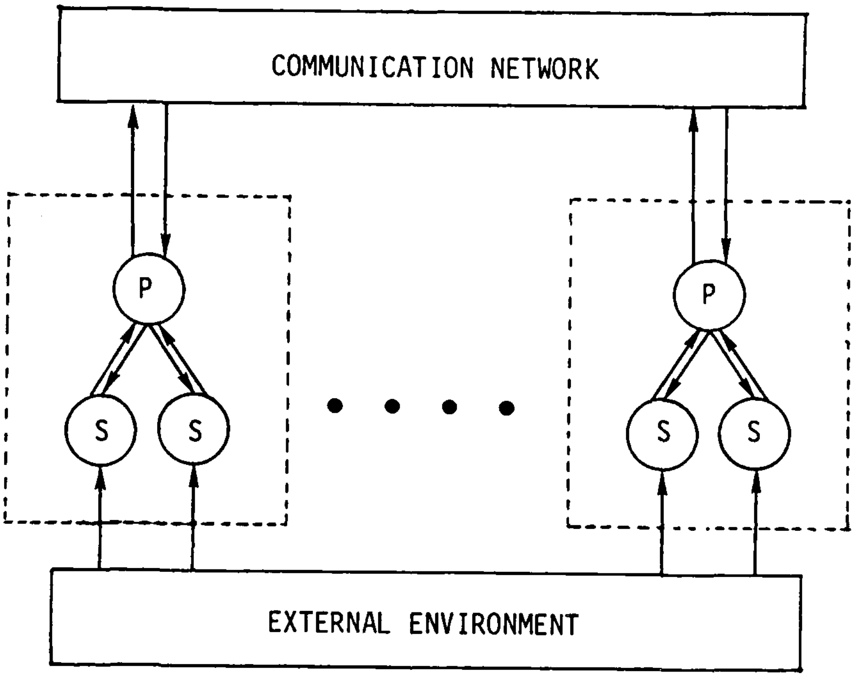

Figure 1. Distributed Estimation System

cheap computing hardware. These advantages make distributed estimation systems extremely attractive for many military and civilian applications. One such application is the distributed sensor network [1], [2] for tracking and surveillance.

Research in distributed estimation has progressed along several directions. A team-theoretic approach has been taken by Barta [3] for decentralized linear estimation and by Tenney and Sandell [4] for distributed detection. Extensions of this work in detection have been made by Teneketzis [5] and Ekchian and Tenney [6]. Another approach, based on finding constrained decentralized filters, has been taken by Tacker and Sanders [7]. The approach of fusion or combining of local estimates to recover the globally optimal estimate has been used in [8] to [12]. The linear problem was considered by Speyer [8], Chong [9], Willsky et al. [10] and Levy et al. [11] while Castanon and Teneketzis [12] considered the nonlinear extension. In all of the above [8]-[11], the system structure is hierarchical with no feedback communication or coordination from the fusion agent. Similar problems of this type have also been considered in the management science literature [13].

The network aspect in the distributed estimation problem has been the emphasis in [14], [15] and discussed in [2]. Borkar and Varaiya [14] presented results on the asymptotic agreement among agents for estimation while Tsitsiklis and Athans [15] considered asymptotic agreement for more general decision problems. It has been demonstrated in [2] via an example that agreement may not be desirable since the common conclusion may be wrong.

In this paper, we elaborate the results obtained in [2]. The philosophy of fusion or combining of local conditional probabilities to obtain the probability conditioned on all available information is again used. However, arbitrary network structures are considered explicitly. They may be hierarchical with or without feedback from the higher level or fully distributed. The presentation is at a fairly elementary level to simplify the notation but can be made more sophisticated if desired by introducing sigma fields. The results may provide the theoretical basis for the analysis and design of systems such as the distributed sensor network.

The rest of this paper is organized as follows. In Section 2, we present the model to be used for distributed estimation. Section 3 describes the distributed estimation problem. Section 4 describes the basic results for static random states. A method for generating the fusion formula for arbitrary networks is given. The fusion algorithms for some examples are also described. Section 5 extends the basic results to the case of dynamic random states. Section 6 is the conclusion.

2. MODEL FOR DISTRIBUTED ESTIMATION

2.1 STATE AND OBSERVATION MODELS

We consider the estimation of a random process $x(t)$, $t \in T$ where $T = [t_o, \infty)$ and $x(t) \in X$. The random process $x(.)$ can be static, deterministic or a general Markov process. We assume the statistics which specify the random process completely are known.

Let S be a finite set of sensors. At a given time t in T, a sensor s generates an output or measurement z in the measurement space Z_s. The triple (z,t,s) is then called a data set and (t,s) is the data set index. Let $\underline{Z}$ be the set of all data sets and $\underline{K}$ be the set of all data set indices. If we assume that each sensor can produce only a finite number of outputs in any finite time interval, the sets $\underline{Z}$ and $\underline{K}$ are at most countable. Furthermore, for each t T, the restrictions

$$Z_{|t} = \{(z,t',s) \in Z | \; t' \leq t\} \qquad (2.1)$$

and

$$K_{|t} = \{(t',s) \in K | \; t' \leq t\} \qquad (2.2)$$

are both finite.

We make two additional assumptions:

1. The sensor origin and time of each data set are known, i.e., for any data set $(z,t,s) \in \underline{Z}$, t and s are known quantities.

2. The measurements are all conditionally independent given the state process, i.e., for any finite subset $\{(z_1,t_1,s_1),\ldots, (z_k,t_k,s_k)\}$ of $\underline{Z}$,

$$\text{Prob.}\left(\bigcap_{i=1}^{k} \{z_i \in dz_i\} | x(t_1),\ldots, x(t_k)\right)$$

$$= \prod_{i=1}^{k} \text{Prob.}(z_i \in dz_i | x(t_i)) \qquad (2.3)$$

With the second assumption, the observation process can be characterized completely by the transition probabilities (or probability densities) from X to Z_s.

2.2 DATA BASES

We are interested in estimation of the process by a network of agents. At any time t, due to communication constraints, each agent may not have access to all available data sets. In general, an agent will have only a subset of the available $Z_{|t}$ at t, corresponding to only a subset of $K_{|t}$. A <u>data base</u> Z at time t is a subset of $Z_{|t}$ and a <u>data index base</u> K at time t is a subset of $K_{|t}$. According to this definition, $Z_{|t}$ ($K_{|t}$) is the maximum data (index) base at t and ϕ (the empty set) is the minimum. Given any data base $Z = \{(z_1,t_1,s_1),\ldots,(z_k,t_k,s_k)\}$, the corresponding data index base $K = \{(t_1,s_1),\ldots,(t_k,s_k)\}$ is found by the operation

$$K = I_n(Z) \tag{2.4}$$

where the definition of I_n is obvious and the actual measurements $(z_1,\ldots,z_k)$ are founa by

$$(z_1,\ldots,z_k) = M_v(Z). \tag{2.5}$$

When $Z = \phi$, $I_n(\phi) = \theta$, and $M_v(\phi) = \theta$ where θ is a symbol representing "no information".

For each data index base $K = \{(t_1,s_1),\ldots,(t_k,s_k)\}$ with corresponding data base $Z = \{(z_1,t_1,s_1),\ldots,(z_k,t_k,s_k)\}$ we define the conditional probability $P(.|Z)$ to mean $P(.|M_v(Z),K)$.

All the definitions above can be given more rigorously in terms of sigma algebras. This will not be attempted in this paper so as to simplify the development.

2.3 COMMUNICATION MODEL

We assume there is a finite set N of estimation agents. Each agent n has its own set of sensors, i.e., a subset S_n of S. Furthermore, the sensor sets are disjoint for different agents, i.e., $S_n \cap S_{n'} = \phi$ for

$n \neq n'$. Each agent n also receives information from other agents via communication. Communication among agents is specified by the known communication schedule $\underline{C}$ which is a subset of T x N x N. $(t,n_1,n_2) \in \underline{C}$ means that agent n_1 transmits some messages to agent n_2 at time t. The exact form of the messages will be discussed later.

Just as in the data set index set, we assume the communication frequency cannot be infinite, so that, for any $t \in T$, the communication schedule up to t,

$$C_{|t} = \{(t',n_1,n_2) \in C | t' \leq t\} \tag{2.6}$$

is finite.

3. DISTRIBUTED ESTIMATION PROBLEM

3.1 INFORMATION GRAPH

The distributed estimation system (N, S, $\underline{C}$) thus consists of the sensor set S and the estimation agent set N together with the communication schedules. Four types of events affect the change of information in the system. These events, the times when they occur and the nodes (sensors or estimation agents) which are affected, are given below:

- sensor observation: $\underline{K}$,
- reception of sensor data by an estimation agent:

  $\{(t,n) \in T \times N | (t,s) \in K, s \in S_n\}$,
- transmission by an estimation agent:

  $\{(t,n) \in T \times N | (t,n,n') \in \underline{C}\}$,
- reception of transmission by an estimation agent:

  $\{(t,n) \in T \times N | (t,n',n) \in \underline{C}\}$.

Consider a subset $\underline{I}$ of T x (S $\cup$ N) which is the union of all the sets defined above. Define an anti-symmetric and transitive binary relation (or partial ordering) $\prec$ on $\underline{I}$ such that

i. For each $(n,t,t') \in N \times T \times T$, $(t,n) \in \underline{I}$, $(t',n) \in \underline{I}$ and $t < t'$ implies that

$$(t,n) \prec (t',n);$$

ii. $(t,s) \in \underline{K}$, $s \in S_n$ and $(t,n) \in \underline{I}$ implies that

$$(t,s) \prec (t,n);$$

iii. $(t,n,n') \in \underline{C}$ implies that

$$(t,n) \prec (t,n').$$

This binary relation or partial order on $\underline{I}$ thus satisfies all the constraints associated with perfect communication as defined by $\underline{C}$ as well as perfect memory at each processing node. $(\underline{I},<)$ characterizes the information flow in the system and is called the <u>information graph</u>. If all the sensor measurements (data sets) can be communicated perfectly through the communication network, the data base Z(t,i) for each node (t,i) in the graph $(\underline{I},\prec)$ can be defined by beginning with the minimal elements and following the rules shown below:

i. If (t,i) is a receiving node,

$$Z(t,i) = \{Z(s,j) \mid (s,j) \rightarrow (t,i)\};$$

ii. If (t,i) is a transmitting node,

$$Z(t,i) = \begin{cases} Z(s,j) & \text{if } (s,j) \rightarrow (t,i) \\ \{(z(k),k)\} & \text{if } (t,i) = k \in \underline{K} \\ \phi & \text{otherwise.} \end{cases}$$

In the above $(s,j) \rightarrow (t,i)$ means that (s,j) is an immediate predecessor of (t,i) and $(z(k),k) \in \underline{Z}$ is the unique element whose second component is k.

With this construction of the data base, we see that $(t,i) \prec (s,j)$ if and only if $Z(t,i) \subseteq Z(s,j)$. Similar remarks can be made for the data index base $K(t,i)$.

Since there is a natural direction (along increasing time) in the graph, the arrowheads on the edges in a pictorial representation of the graph can be omitted. We would also omit those edges which are due to transitivity. From the graph, the flow of information in the system becomes very obvious. A node (t,i) is a parent of (s,j) if information flows from node i at time t to node j at time s. Note that in the information graph, the receiving nodes correspond to the events when estimates have to be updated with the arrival of new information. For many applications, it is sufficient to use a reduced information graph, which is obtained by considering only these receiving nodes.

Several examples of distributed estimation networks and their information graphs are shown below.

Example 1: (Fusion Without Coordination)

Of the agents in N, agent 1 is a fusion agent and the rest are local agents. The local agents transmit to the fusion agent after they receive the data from the sensors and perform local processing. Figure 2 shows the structure of the system (for three agents) and the information graphs. In this case

$$N = \{1,\ 2,\ 3\}$$

$$\underline{C} = \bigcup_{i=1}^{\infty} \{(s_i,2,1),(s_i,3,1)\}$$

where $\{s_0 < s_1 < s_2 < \ldots\}$ are the communication times, and $\{t_0 < t_1 < t_2 < \ldots\}$ are the sensor observation times.

Example 2: (Fusion With Coordination)

This is similar to Example 1 except that right after fusion, Agent 1 communicates with the local agents again. This structure is also equivalent to a broadcast system where all agents communicate with each other. For $N = \{1,\ 2,\ 3\}$, the communication schedule is given by

$$\underline{C} = \bigcup_{i=1}^{\infty} \bigcup_{n_1 \neq n_2} \{(s_i,n_1,n_2)\}$$

where $s_0 < s_1 < s_2 < \ldots\ldots$. Figure 3 shows the structure of the system and the information graphs.

Example 3: (Cyclic Communication)

This is the example considered in [2]. The agents are arranged in a circle as in Figure 4. Each agent transmits only to its immediate neighbor in a cyclic manner at the specified communication times. Figure 4 shows the example for $N = \{1,\ 2,\ 3\}$

$$\underline{C} = \bigcup_{i=1}^{\infty} \{(s_i,1,3),\ (s_i,3,2),\ (s_i,2,1)\}$$

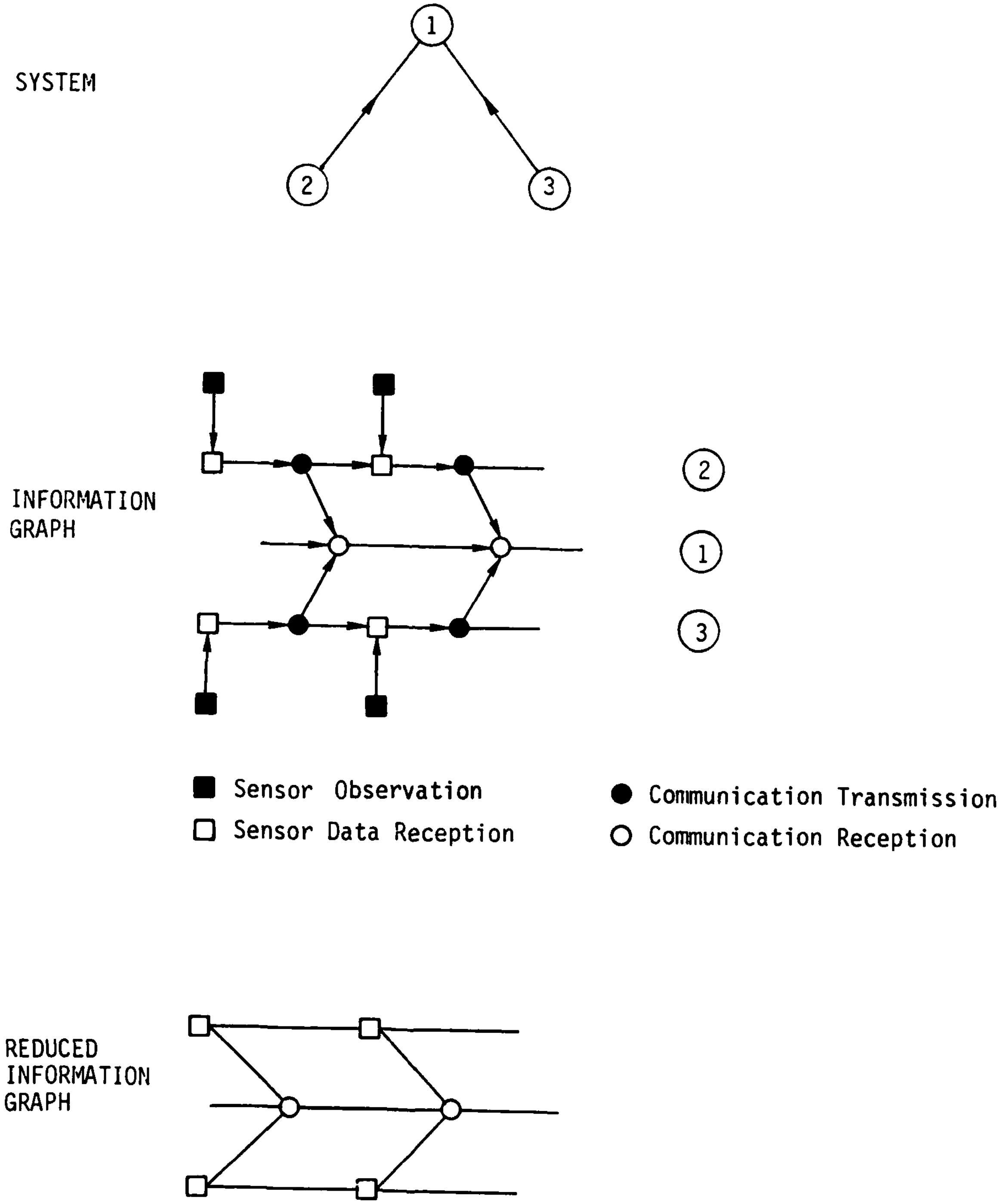
SYSTEM
1
2
3
INFORMATION GRAPH
2
1
3
■ Sensor Observation
□ Sensor Data Reception
● Communication Transmission
○ Communication Reception
REDUCED INFORMATION GRAPH

Figure 2. Fusion Without Coordination

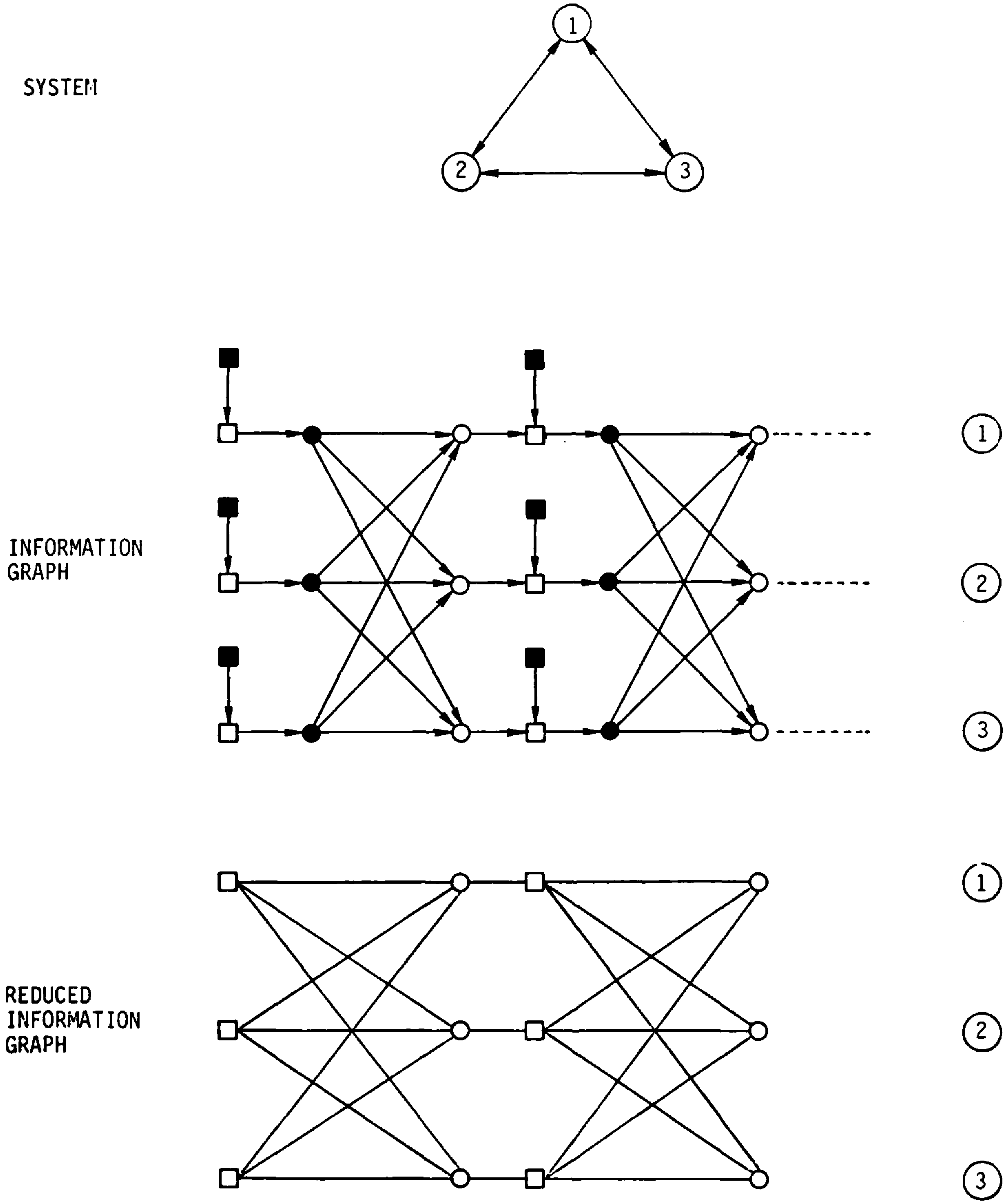

Figure 3. Fusion With Coordination

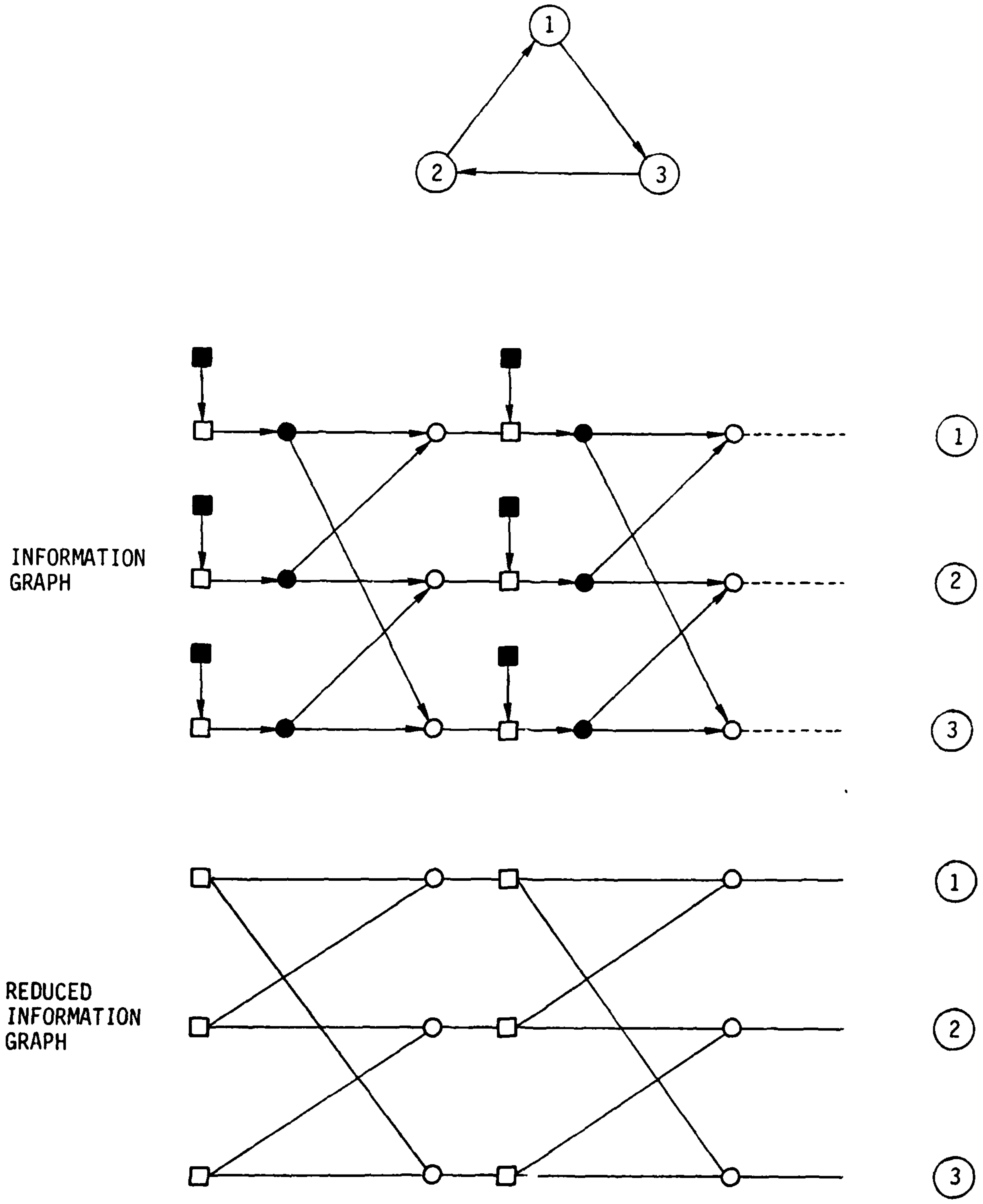

Figure 4. Cyclic Communication

with $s_0 < s_1 < s_2 < \ldots\ldots$

Example 4: (Multipath Pattern)

The agents are arranged as in Figure 5. Agent 1 can only get information from Agent 4 via Agents 2 and 3. For N = {1,2,3,4} and

$$\underline{C} = \bigcup_{i=1}^{\infty} \{(s_i,2,1),\ (s_i,3,1),\ (s_i,4,2),\ (s_i,4,3)\},$$

the information graphs are given in Figure 5.

3.2 DISTRIBUTED FUSION PROBLEM

The problem is to compute $p(x(t)|Z(t,i))$ for each node $(t,i) \in \underline{I}$ in the graph $(\underline{I}, \prec)$. Since the conditional probabilities or any estimates are updated only at the receiving nodes (extrapolation is carried out at the other nodes), we need only to consider the computations at the following two types of nodes in the reduced information graph:

- a sensor data reception node,
- a communication reception node.

At a sensor data reception node (t,i), computation of $p(x(t)|Z(t,i))$ is straightforward. The standard Bayesian update formula would suffice. At a communication reception node, the objective is to

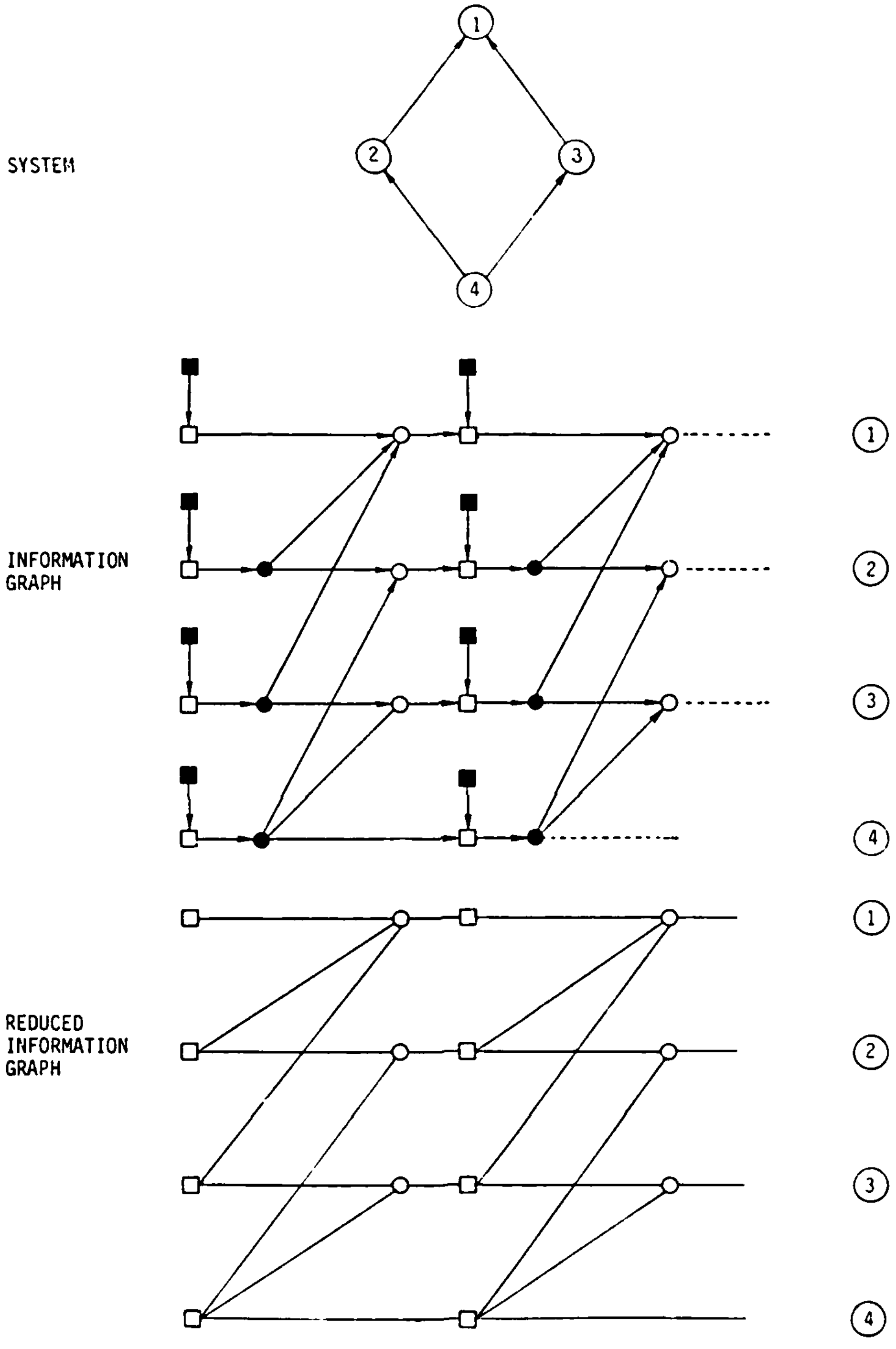

Figure 5. Multipath Pattern

reconstruct $p(x(t)|Z(t,i))$ from the conditional probabilities $\{p(x(t)|Z(s,j))|(s,j) \prec (t,i)\}$. This problem is the distributed fusion problem: construction of the conditional probability given all the data sets which would have been communicated through the network using only the conditional probabilities available at the predecessor nodes in the information graph.

4. STATIC RESULTS

In this section we develop the main results for fusion for each agent i, assuming the random process is static, i.e., x(t) = x for all t. Since the information from different agents may overlap, care has to be taken when the conditional probabilities from different agents are combined. In particular, any redundant information has to be identified so that it is not used more than once. The following lemmas provide the mechanism for doing this. In the following x denotes a random vector with prior probability p(x) and $\underline{Z}$ is the set of all data sets.

4.1 BASIC RESULTS

Lemma 1

Suppose Z_1 and Z_2 are data bases at two information nodes 1 and 2. Then

$$p(x|Z_1 \cup Z_2) = C \frac{p(x|Z_1)\ p(x|Z_2)}{p(x|Z_1 \cap Z_2)} \qquad (4.1)$$

where C is a normalization constant.

Proof

In the following $A \setminus B$ denotes the difference of two sets. By Bayes´ rule,

$$p(x|Z_1 \cup Z_2) = \frac{p(Z_1 \cup Z_2|x)\ p(x)}{p(Z_1 \cup Z_2)}. \tag{4.2}$$

Since $Z_1 \cup Z_2$ can be written as

$$Z_1 \cup Z_2 = (Z_1 \setminus Z_2) \cup (Z_2 \setminus Z_1) \cup (Z_1 \cap Z_2), \tag{4.3}$$

where the three disjoint data bases are conditionally independent given x,

$$p(x|Z_1 \cup Z_2)$$

$$= \frac{p(Z_1 \setminus Z_2|x)\ p(Z_2 \setminus Z_1|x)\ p(Z_1 \cap Z_2|x)\ p(x)}{p(Z_1 \cup Z_2)}. \tag{4.4}$$

But

$$p(Z_1|x) = p(Z_1 \setminus Z_2|x)\ p(Z_1 \cap Z_2|x),$$

$$p(Z_2|x) = p(Z_2 \setminus Z_1|x)\ p(Z_1 \cap Z_2|x). \tag{4.5}$$

Thus,

$$p(x|z_1 \cup Z_2) = \frac{p(Z_1|x)\ p(Z_2|x)\ p(x)}{p(Z_1 \cap Z_2|x)\ p(Z_1 \cup Z_2)}, \tag{4.6}$$

which reduces to (4.1).

This lemma states that since $p(x|Z_1)$ and $p(x|Z_2)$ both include information contained in the data base $Z_1 \cap Z_2$, this common information has to be removed so that it does not get double counted. Lemma 1 plays a central role in distributed estimation theory similar to the usual Bayes´ rule in centralized estimation theory.

Lemma 2, which is a special case of Lemma 1, is also quite useful.

<u>Lemma 2</u>

Suppose $Z_1 \cap Z_2 = \phi$. Then

$$p(x|Z_1 \cup Z_2) = C \frac{p(x|Z_1)\ p(x|Z_2)}{p(x)}. \tag{4.7}$$

When the conditional probabilities from multiple agents are combined, the fusion formula can be obtained by repeated applications of Lemma 2. The following gives the results for three agents.

<u>Lemma 3</u>

Suppose Z_1, Z_2 and Z_3 are data bases at the information nodes 1, 2 and 3. Then

$$p(x|Z_1 \cup Z_2 \cup Z_3) = C \frac{p(x|Z_1 \cup Z_2)\ p(x|Z_3)}{p(x|(Z_1 \cup Z_2) \cap Z_3)}$$

$$= C \frac{p(x|Z_1)\ p(x|Z_2)\ p(x|Z_3)\ p(x|Z_1 \cap Z_2 \cap Z_3)}{p(x|Z_1 \cap Z_2)\ p(x|Z_2 \cap Z_3)\ p(x|Z_3 \cap Z_1)}, \tag{4.8}$$

This lemma again has a very intuitive explanation. The terms in the denominator consist of pairwise redundant information to be removed. When these are removed, all information which is common to Z_1, Z_2, and Z_3 is also removed. This then has to be restored.

If all the random elements involved are Gaussian, the lemmas above can be simplified so that only the conditional means and covariances are involved. Suppose x is Gaussian with mean m and covariance P(0). Let $\hat{x}(Y)$ and P(Y) be the mean and covariance corresponding to the conditional density $p(x|Y)$. Then lemma 1 becomes

<u>Lemma 1A</u>

$$P(Z_1 \cup Z_2)^{-1} = P(Z_1)^{-1} + P(Z_2)^{-1} - P(Z_1 \cap Z_2)^{-1} \tag{4.9}$$

and

$$P(Z_1 \cup Z_2)^{-1} \hat{x}(Z_1 \cup Z_2)$$
$$= P(Z_1)^{-1}\hat{x}(Z_1) + P(Z_2)^{-1}\hat{x}(Z_2) - P(Z_1 \cap Z_2)^{-1}\hat{x}(Z_1 \cap Z_2). \tag{4.10}$$

Lemma 2 and Lemma 3 can be simplified in a similar way. Lemma 1A is identical to that used in [9] for deriving the optimal algorithms for combining estimates of linear Gaussian systems.

We now state the static fusion problem for each agent assuming that $x(t) = x$ for all t. The problem is stated for the case when messages are received from only one agent. But the extension to multiple agents is obvious.

Static Fusion Problem

Suppose agent i receives a message from agent j at time s in the form of a conditional probability $p(x|Z(s,j))$. Let (t,i) be the immediate predecessor to (s,i) for agent i. Agent i's data base then changes from $Z(t,i)$ to

$$Z(s,i) = Z(t,i) \cup Z(r,j). \tag{4.11}$$

where (r,j) is the immediate predecessor to (s,j) for agent j. The objective is to find $p(x|Z(s,i))$ in terms of $p(x|Z(t,i))$, $p(x|Z(r,j))$ and possibly other conditional probabilities defined on the information graph, i.e., $\{p(x|Z(t',i'))|(t',i') \prec (s,i)\}$.

We do not specify a priori which conditional probabilities are involved except they have to be conditional on some data base $\underline{Z}$ defined on the information graph and that they should be available through communication. The following recursive algorithm allows us to find the set of needed conditional probabilities and how they should be combined.

Algorithm for Static State

The algorithm consists of repeated applications of the following steps.

Step 1

Since Z(t,i) and Z(r,j) are subsets of $\underline{Z}$, Lemma 1 gives

$$p(x|Z(s,i)) = p(x|Z(t,i) \cup Z(r,j))$$

$$= C \frac{p(x|Z(t,i))\, p(x|Z(r,j))}{p(x|Z(t,i) \cap Z(r,j))} \qquad (4.12)$$

If $Z(t,i) \cap Z(r,j)$ is the data base for some node in the information graph, i.e., $Z(t,i) \cap Z(r,j) = Z(q,k)$ for some (q,k) in $\underline{I}$ or if it is empty, then the algorithm terminates. If not, Step 2 is used. In terms of the information graph representation introduced in Section 3, this step is particularly simple. We start from two information nodes (t,i) and (r,j) whose conaitional probabilities are to be combined. Z(t,i) Z(r,j) corresponds to the information of all those nodes which are parents of both (t,i) and (r,j).

<u>Step 2</u>

Let $\{(t_1,k_1), (t_2,k_2),\ldots\}$ be the set of commcn predecessors of (t,i) and (r,j) in the information graph. Then

$$Z(t,i) \cap Z(r,j) = Z(t_1,k_1) \cup Z(t_2,k_2) \cup \ldots \qquad (4.13)$$

Step 1 can now be repeated with the help of Lemma 1 (and its multiple agent version) to express $p(x|Z(t,i) \cap Z(r,j))$ in terms of the conditional probabilities $p(x|Z(t_i,k_i))$, i = 1, 2,..., and $p(x|Z(t_i,k_i) \cap Z(t_j,k_j))$, i = 1, 2,..., j = 1, 2, ..., etc. The algorithm terminates when all the conditional probabilities are defined on nodes in the information graph or coincide with the *a priori* distributions.

By applying this algorithm, $p(x|Z(t,i) \cup Z(r,j))$ can be expressed in terms of products and ratios of conditional probabilities defined on information nodes. Each product corresponds to the fusion or combining of information whereas each division corresponds to the removal of redundant information. Note that in general it is not sufficient to use only the conditional probabilities $p(x|Z(t,i))$ and $p(x|Z(r,j))$ unless $Z(t,i)$ and $Z(r,j)$ happen to be disjoint or there is a node (s,k) such that $Z(s,k) = Z(t,i) \cap Z(r,j)$. Additional conditional probabilities from the past are also neeaed so that the redundant information in $Z(t,i)$ and $Z(r,j)$ can be identified and removed. Two cases are possible.

<u>Case 1</u>: The additional conditional probabilities are all available to agent i, i.e., they are either generated locally from measurements, or they are received from other agents.

<u>Case 2</u>: The additional conditional probabilities may not be available to agent i. In this case, additional communication may be added. However, from the algorithm, it can be seen that existing communication paths are available to pass along these conditional probabilities.

We have thus solved the fusion problem for each agent in a distributed estimation network. This algorithm also provides us with the set of conditional probabilities which needs to be stored at each agent plus the additional set of conditional probabilities which needs to be communicated.

When the random elements involved are all Gaussian, the sufficient statistics for the conditional probabilities become the conditional means and covariances. With the help of Lemma 1A, we can again apply the algorithm. Instead of multiplication and division of probabilities, however, we now have operations involving conditional means and covariances. The results are straightforward and will not be presented here.

4.2 STATIC EXAMPLES

In the following we assume the measurements are made at times {..., t-1, t+1,...} and messages are received at times {..., s-1, s+1,...} with s-1 < t < s.

Example 1: (Fusion Without Coordination)

Consider the fusion time s. Let t be the observation time immediately before s. With the information graph it is easy to see that

$$Z(s-1,1) \cap Z(t,2) = Z(t-1,2). \tag{4.14}$$

Thus

$$p(x|Z(s-1,1) \cup Z(t,2)) = C\,\frac{p(x|Z(t,2))\;p(x|Z(s-1,1))}{p(x|Z(t-1,2))}$$

By a recursive argument, we can show that

$$p(x|Z(s,1)) = C \prod_{i\neq 1} \frac{p(x|Z(t,i))}{p(x|Z(t-1,i))} \; p(x|Z(s-1,1)). \tag{4.15}$$

Each term in the product contains the new information contained in the new measurement z(t,i) of agent i. All other information is already known to agent 1. The fusion problems of the other agents are similar.

<u>Example 2</u>: (Fusion with Coordination or Broadcast System) From the information graph, we see that for all j,

$$\bigcap_{i} Z(t,i) = Z(s-1,j). \tag{4.16}$$

Thus, the algorithm gives for j,

$$p(x|Z(s,j)) = C \prod_{i} \frac{p(x|Z(t,i))}{p(x|Z(s-1,i))} \; p(x|Z(s-1,j)). \tag{4.17}$$

Each term in the product is the new information contained in measurement z(t,i).

<u>Example 3</u>: (Cyclic Communication)

$$Z(t,1) \cap Z(t,2) = Z(t-2,1) \cup Z(t-1,2), \tag{4.18}$$

and

$$\begin{aligned} Z(t-2,1) \cap Z(t-1,2) &= Z(s-3,1) \\ &= Z(t-3,1) \cup Z(t-3,2). \end{aligned} \tag{4.19}$$

Thus, the algorithm gives for general i = 1,2,3

$$p(x|Z(s,i)) = C \frac{p(x|Z(t,i))}{p(x|Z(t-2,i))} \frac{p(x|Z(t,[i+1]))}{p(x|Z(t-1,[i+1]))} p(x|Z(s-3,i)) \qquad (4.20)$$

where [i] is i modulo 3.

Thus, in addition to the most current conditional probability $p(x|Z(t,1))$, agent 1 has to remember three other probabilities. Note that $p(x|Z(t-1,2))$ is available to agent 1 from earlier communications. This indicates that in a distributed estimation network, knowing the most recent estimate is frequently not sufficient if one wants to recover the globally optimal estimate. In fact, it has been shown via simulation in [2] that if a suboptimal rule of combining estimates is used, such as

$$p(x|Z(t,1) \cup Z(t,2)) \approx C\ p(x|Z(t,1))\ p(x|Z(t,2)) \qquad (4.21)$$

for agent 1 and similar rules for agents 2 and 3, the agents agree asymptotically. This is consistent with the results on asymptotic agreement in distributed estimation as given in [14]. However, the agents can converge to the wrong estimate as demonstrated in [2]. Thus, although optimal fusion algorithms are in general more complicated, requiring more memory and more computation, they are nonetheless necessary if good performance is needed. A suboptimal algorithm has also been tested in [2] and shown to have some nice properties.

Example 4 (Multipath Pattern)

The fusion problems of agents 2 and 3 are straightforward. For agent 1, repeated use of the algorithm (with the help of the information graph in Figure 5) gives

$p(x|Z(s,1))$ =

$$C \frac{p(x|Z(t,2))}{p(x|Z(t-1,2))} \frac{p(x|Z(t,3))}{p(x|Z(t-1,3))} \frac{p(x|Z(t-2,4))}{p(x|Z(t-1,4))} p(x|Z(t,1)) \qquad (4.22)$$

In addition to the conditional probabilities from agents 2 and 3, conditional probabilities by agent 4 are also needed. These would have to be relayed by agents 2 or 3.

In the above examples, general fusion formulas are given. If the random vectors are all Gaussian, these formulas can be simplified using Lemma 1A.

5. DYNAMIC RESULTS

Assume now that x(.) is a Markov process. The fusion problem for each agent will now be considereo. Since the data sets are no longer conditionally independent given x(t), one immediate question is the choice of an appropriate "state" y whose conditional probabilities would be computed, transmitted and combined by the various agents. Let T(t,i) be

$$T(t,i) = \{t' \in T \mid (t',i') \in K(t,i)\}, \tag{5.1}$$

and

$$y = (x(t'))_{t' \in T(t,i)} \tag{5.2}$$

for each information node (t,i) where fusion is to be performed. Then the problem is effectively reduced to a static problem of the type considered in Section 4. Using the independence assumptions on the measurements in the data base given y, the algorithm in Section 4 can be applied. However, this means that the conditional probability of a high dimensional random vector y would have to be stored and transmitted. From an implementational point of view, this may not be feasible.

For deterministic random processes, which can be characterized by the state at one given time, an obvious choice is to estimate $x(t_o)$ where t_o is the minimum in the set T. Again, due to the Markov property, the conditional independence assumption is satisfied and the algorithm can be used. However, if there are substantial changes in the process, $x(t_o)$ may not be the state of interest. In this section, we characterize the more current states whose conditional probabilities ought to be transmitted and combined.

The following generalization of Lemma 1 is needed.

Lemma 4

Consider a random vector y and data bases Z_1 and Z_2 defined on the information graph. Suppose

$$p(Z_1 \setminus Z_2, Z_2 \setminus Z_1 | Z_1 \cap Z_2, y)$$

$$= p(Z_1 \setminus Z_2 | y, Z_1 \cap Z_2)\ p(Z_2 \setminus Z_1 | y, Z_1 \cap Z_2). \tag{5.3}$$

Then

$$p(y|Z_1 \cup Z_2) = C \frac{p(y|Z_1)\ p(y|Z_2)}{p(y|Z_1 \cap Z_2)} \tag{5.4}$$

where C is a normalization constant.

Proof

By Bayes´ rule and (5.3), we have

$$p(Z_1 \cup Z_2, y) = p(Z_1 \setminus Z_2, Z_2 \setminus Z_1, Z_1 \cap Z_2, y)$$

$$= p(Z_1 \setminus Z_2, Z_2 \setminus Z_1 | Z_1 \cap Z_2, y) p(Z_1 \cap Z_2, y)$$

$$= p(Z_1 \setminus Z_2 | Z_1 \cap Z_2, y) p(Z_2 \setminus Z_1 | Z_1 \cap Z_2, y) p(Z_1 \cap Z_2, y)$$

$$= \frac{p(Z_1 \setminus Z_2, Z_1 \cap Z_2, y) p(Z_2 \setminus Z_1, Z_1 \cap Z_2, y)}{p(Z_1 \cap Z_2, y)}$$

$$= \frac{p(Z_1, y)\ p(Z_2, y)}{p(Z_1 \cap Z_2, y)}. \tag{5.5}$$

The lemma then follows naturally.

Lemma 4 states that even though the individual measurements in $\underline{Z}$ do not satisfy the conaitional independent assumptions given y, Equation (5.4) (which is the same as (4.1)) is still valid provided the private data bases $Z_1 \setminus Z_2$, $Z_2 \setminus Z_1$ are conditionally independent given the state y and the common information $Z_1 \cap Z_2$.

We can now state the following theorem which characterizes the state vector which should be estimated for deterministic dynamic random processes.

<u>Theorem</u>

Consider the fusion problem for the information node (t,i) assuming a deterministic random process x. If the algorithm of Section 4 yields the fusion formula

$$p(x|Z(t,i)) = F(p(x|Z(t',i'));(t',i') \in L(t,i)) \tag{5.6}$$

where F is a function consisting of products and ratios of $p(x|Z(t',i'))$'s in the set $L(t,i)$, and $L(t,i)$ is a subset of the predecessor information nodes of (t,i). Then for a deterministic random process $x(.)$, equation (5.6) holds with x replaced by $x(t^*)$, where

$$t^* = \min\{t' | (t,s) \in L(t,i) \backslash \{(t'',s'')\}\} \tag{5.7}$$

and (t'',s) is the minimal element in $L(t,i)$.

The proof is straightforward and is based on the algorithm of Section 4 and Lemma 4.

This theorem states that for random processes, in general the filtered estimate represented by the conditional probabilities $p(x(t)|Z(t,i))$ may not be adequate for optimal fusion at time t. Sometimes the agents need to have the conditional probabilities of the states at some earlier times. Thus, smoothed estimates are frequently needed. From this, the estimates of the current states can be obtained easily by extrapolation, e.g.,

$$p(x(t)|Z(t,i)) = \int p(x(t)|x(s))\, p(x(s)|Z(t,i))\, dx(s). \tag{5.8}$$

When this theorem is applied to the examples in Section 4, we obtain the following results.

<u>Example 1</u>: (Fusion without Coordination)

In the fusion equation (4.15), the state to be estimated is x(t). This is consistent with the results in [8]-[12].

As a variation of this, consider a periodic fusion situation where the local agents acquire measurements at a higher rate than they communicate with the fusion agent (Figure 6). Specifically, let the new fusion time set for agent 1 be {...,s-M,s,s+M,..,}where M is the number of time units between communication.

Then application of the theorem yields

$$p(x(t-M+1)|Z(s,1)) = C \prod_{i \neq 1} \frac{p(x(t-M+1)|Z(t,i))}{p(x(t-M+1)|Z(t-M,i))}$$

$$p(x(t-M+1)|Z(s-M,1)) \qquad (5.9)$$

Thus, the state of interest is now x(t-M-1), and each term in the product contains the new information of agent i about this state.

<u>Example 2</u>: (Fusion with Coordination)

In equation (4.17), the state is x(t).

<u>Example 3</u>: (Cyclic Communication)

In equation (4.18), the state is x(t-2). Thus, extrapolation is needed if the estimate of x(t) is needed.

<u>Example 4</u>: (Multipath Pattern)

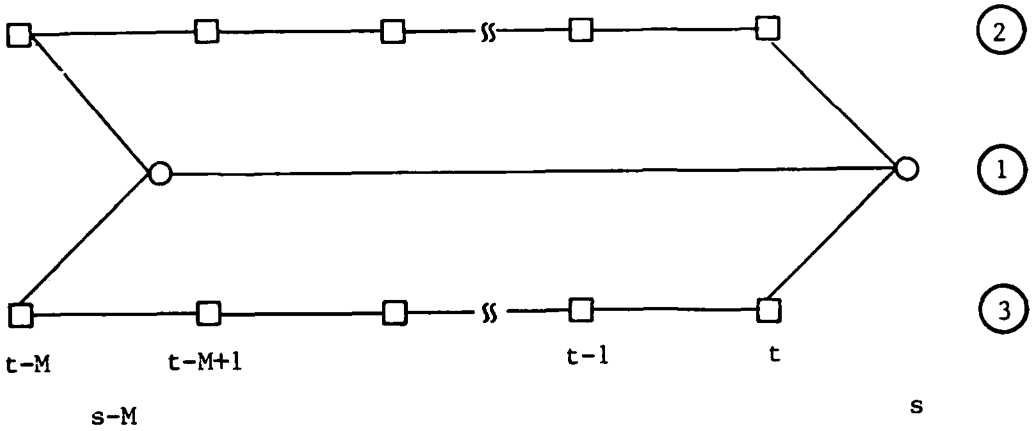

Figure 6. Periodic Fusion Without Coordination

In equation (4.22), the state is x(t-1). Thus, extrapolation is again needed.

6. CONCLUSION

We have presented a formalism for the distributed estimation problem. Using this formalism, the optimal fusion algorithm for each agent in the network has been developed for arbitrary network structures. Both results for static and deterministic dynamic random states have been described, and illustrated with examples.

The results have been presented for very general state and observation models. Special cases such as linear models with Gaussian noises can be considered. An interesting special case for distributed multi-target tracking and classification has also been investigated and briefly reported in [2]. The details will appear elsewhere.

REFERENCES

1. R. Lacoss, "Overview of the Distributed Sensor Networks Program at Lincoln Laboratory," *Proc. Distributed Sensor Networks Workshop*, MIT Lincoln Lab. Jan. 6-7, 1982.

2. C.Y. Chong, S. Mori, E. Tse and R.P. Wishner, "Distributed Estimation in Distributed Sensor Networks," *Proc. 1982 American Control Conference*, Arlington, VA, 1982.

3. S.M. Barta, "On Linear Control of Decentralized Stochastic Systems," Ph.D. Dissertation, MIT, Cambridge, MA, July 1978.

4. R.R. Tenney and N.R. Sandell, Jr., "Detection with Distributed Sensors," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-17, No. 4, pp. 501-510, July 1981.

5. D. Teneketzis, "The Decentralized Quickest Detection Problem," *Proc. 21st IEEE Conf. on Decision and Control*, Fort Lauderdale, FL, 1982.

6. L.K. Ekchian and R.R. Tenney, "Detection Networks," *Proc. 21st IEEE Conf. on Decision and Control*, Ft. Lauderdale, FL, 1982.

7. E.C Tacker and C.W. Sanders, "Decentralized Structures for State Estimation in Large Scale Systems," *Large Scale Systems Theory and Applications*, Vol. 1, No. 1, pp. 39-49, Feb. 1980.

8. J.L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," *IEEE Trans. Automat. Contr.*, Vol. AC-24. No. 2, pp. 266-269, April 1979.

9. C.Y. Chong, "Hierarchical Estimation," *Proc. 2nd MIT/ONR Workshop on Distributed Communication and Decision Problems*, July 1979.

10. A.S. Willsky, M. Bello, D.A. Castanon, B.C. Levy and G. Verghese, "Combining and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks," *IEEE Trans Automat. Contr.*, Vol. AC-27, No. 4, pp. 799-813, Aug. 1982.

11. B.C. Levy, D.A. Castanon, G.C. Verghese and A.S. Willsky, "A Scattering Framework for Decentralized Estimation Problems," *Automatica*, vol. 19, no. 4, pp. 373-394, July, 1983.

12. D.A. Castanon and D. Teneketzis, "Nonlinear Data Fusion," *Proc. 5th MIT/ONR Workshop on* C^3 *Systems*, Monterey, CA, 1982.

13. R.L. Winkler, "Combining Probability Distributions from Dependent Information Sources," *Management Science*, Vol. 27, No. 4, pp.

479-488, April 1981.

14. V. Borkar and P. Varaiya, "Asymptotic Agreement in Distributed Estimation," IEEE Trans. Automat. Contr., Vol. AC-27, No. 3, pp. 650-655, June 1982.

15. J.N. Tsitsiklis and M. Athans, "Convergence and Asymptotic Agreement in Distributed Decision Problems," Proc. 21st IEEE Conf. on Decision and Control, Ft. Lauderdale, FL, 1982.

APPENDIX C

DISTRIBUTED MULTITARGET TRACKING AND CLASSIFICATION - A BAYESIAN APPROACH

C.Y. Chong and S. Mori

ABSTRACT

The tracking and classification of multiple targets by a network of processing agents (processors) is considered. A Bayesian approach is adopted as the theoretical basis. Each agent processes the local sensor data to obtain the local information state consisting of the local hypothesis, tracks and the relevant probabilities and state distributions. These are communicated to the other agents by means of the communication network. From these, each agent tries to construct the global information state conditioned on the data which would be available if they were communicated through the network. Both results for static and dynamic target models are presented assuming broadcast type communication.

1. INTRODUCTION

The tracking and classification of multiple targets is very important for many civilian and military applications. It is also interesting from a theoretical standpoint since it is essentially different from classical estimation problems in that the origins of the measurements are uncertain. Many algorithms for multi-target tracking have been proposed. Surveys of the area can be found in the paper by Bar-Shalom [1] and the Naval Ocean Surveillance Correlation Handbooks [2], [3]. The paper by Reid [4] also contains a good survey of the then existing methods. Recently, a general theory for the tracking and classification of multiple targets based on a Bayesian approach has been proposed in [5] and [6]. Much of the work, however, assumes a centralized processing architecture in that the sensor measurements are transmitted to a single processor where they are processed.

In many applications, however, the sensor measurements are not all transmitted to a central processor. Instead, a set of local processors are present and each processor handles the measurements from a subset of the sensors. Each processor does some local tracking and communicates the results to other processors where the incoming information is combined or fused with the local information. Such architectures are present whenever tracking is carried out by multiple processors who communicate. The distributed sensor network is an example of such systems [7], [8].

In recent years, there has been growing interest in distributed estimation problems [9] - [21]. Most of the work deals with the estimation of a random process or hypothesis testing assuming the origins of measurements are known. Exceptions can be found in [22], [23] which presents some ad hoc schemes for distributed multitarget tracking and [8], which briefly outlines some theoretic results. Some specific results have also been considered in [24], [25] which consider the problem of correlation of tracks from multiple nodes. This work is, however, quite ad hoc and not related to any theory of multitarget tracking.

In this paper we present a theory for distributed multitarget tracking and classification assuming the independent and identically distributed target models of [5] or [6]. Each processor forms the data-association hypotheses, tracks and the various associated probabilities and communicates these to the other processors through the network. Upon receiving these, each node tries to reconstruct the global hypotheses, tracks, probabilities of hypotheses and the state distributions of the tracks as if the sensor measurements were available through the network. The theoretical framework introduced in [21] for distributed estimation and the theory of multitarget tracking of [5], [6] are used to derive the fusion algorithms for each processor in the network. Although the philosophy can be used for general communication structures, the special case of broadcast communication has been used to illustrate the algorithm.

The structure of this paper is as follows. In Section 2, we present the basic target and sensor models used. The information structure of the system, which depends on the communication network, is also introduced. Section 3 deals with the notions of tracks and hypotheses and defines the distributed multitarget tracking and classification problem. The main results for tracking of stationary targets assuming broadcast type communication are described in Section 4. In particular we discuss the construction of the global hypotheses from the local hypotheses and the hypothesis evaluation problem. The extension of these results to dynamic target models is given in Section 5. Section 6 contains the conclusion.

2. MODELS

The main difference between distributed multitarget tracking and centralized multitarget tracking is in the presence of multiple tracking agents. Thus the target and sensor models would be identical to the centralized case [5], [6], but additional constraints or models would describe the information available to each node, i.e., the information structure. In the following we shall discuss the three models separately.

2.1 TARGET MODEL

A general target model used in multitarget tracking and classification has been described in [5], [6]. Although a theory of distributed multitarget tracking and classification can be developed using general models, our emphasis in this paper is on a special but widely applicable target model, namely, independent and identically distributed (i.i.d.) target models. Specifically, the target system state at any time t is

$$((x_i(t))_{i=1}^{N_T}, N_T) \tag{2.1}$$

where N_T is the constant but unknown number of targets and $x_i(t)$ is the state of the i^{th} target at time t. The a priori distribution of N_T is Poisson with mean ν_0. Given N_T, $(x_i(t))_{i=1}^{N_T}$ is a system of independent and identically distributed Markov processes on the common target state space X. Each $x_i(.)$ has the same initial distribution/density q_0 and the transition probability density f, i.e.,

$$\text{Prob.}\{x_i(t_0) \in dx\} = q_0(x)\mu(dx) \tag{2.2}$$

$$\text{Prob.}\{x_i(t + \Delta t) \in dx | x_i(t) = x'\} = f_{\Delta t}(x|x')\mu(dx) \tag{2.3}$$

where μ is the hybrid measure on X which is a hybrid set. A hybrid set is the direct product of a subset of Euclidean space and a finite set and a hybrid measure is the direct product measure of the Lebesgue and counting (discrete) measures. Thus each $x_i(t)$ consists of a continuous part corresponding to position, velocity, etc., and a discrete part corresponding to target type, maneuvering mode, etc.

2.2 SENSOR MODEL

We assume there is a system of sensors called S. For each sensor s in S, the sensor output space Z_s is

$$Z_s = \bigcup_{m=0}^{\infty} (Y_s)^m \times \{m\} \tag{2.4}$$

where Y_s is the measurement space for sensor s. Each sensor output $((y_i)_{i=1}^{m}, m)$ means that m measurements are generated and the measurement values are $y_1, \ldots, y_m$. in general, Y_s is a hybrid set, where the continuous part is used for analog information such as position and velocity and the discrete part is used for feature-type information such as size/cross-section, classification, etc.

A Random Element

$$((y_j)_{j=1}^{N_M}, N_M, t, s) \in \bigcup_{s \in S} Z_s \times T \times \{s\} \qquad (2.5)$$

is called a data set. It represents the event that N_M measurements $y_1, \ldots, y_{N_M}$ are generated by sensor s at time t. Given a target system state $((x_i(t))_{i=1}^{N_T}, N_T)$, the data set is generated via the following four steps:

a. Detection

Let $I_T = \{1, \ldots, N_T\}$ be the set of target indices. Then the set of targets detected by a sensor s at time t is a random subset $I_D(t,s)$ of I_T which can be characterized by its indicator function $F_D(t,s)$. $F_D(t,s)$ which is a random binary function with domain I_T, is called the detection function. $F_D(t,s)(i) = 1$ means that target i is detected by sensor s at time t and 0 means that it is not detected. We assume that every $F_D(t,s)(i)$ depends only on target i's state $x_i(t)$ and that there exists a common detection probability function $p_D(x|t,s)$ such that $\text{Prob.}\{F_D(t,s)(i)=1|x_i(t),N_T\} = p_D(x_i(t)|t,s)$.

b. Number of False Alarms Generation

The number of false alarms, $N_{FA}(t,s)$, generated by sensor s at time t depends only on t and s. Specifically,

$$\text{Prob.}\{N_{FA}(t,s) = m|F_D(t.s),\ (x_i(t))_{i=1}^{N_T},\ N_T\} = p_{N_{FA}}(m|t,s) \quad (2.6)$$

The total number of measurements in the data set is then given as

$$N_M(t,s) = N_D(t,s) + N_{FA}(t,s) \quad (2.7)$$

where $N_D(t,s)$ is the number of detected targets.

c. Random Assignment

We assume that the order in which the measurements arrive in a data set does not contain any information about the targets. If not, the data set should be further subdivided until this assumption holds. Let A(t,s) be the random assignment function which assigns the detected targets to the measurements. Since the order of the measurements does not contain any useful information, the probability of A(t,s) taking on any possible assignment α is uniform.

d. Measurement Value Generation

The value of a false alarm is an independent random variable (vector) and has a common probability distribution/density $p_{FA}(y_j|t,s)$. For any detected target x_i, given an assignment $A(t,s) = \alpha$, the corresponding measurement value $y_{\alpha(i)}$ is an independent random vector with a transition probability density

$$p_m(y_{\alpha(i)} | x_i, t, s). \quad (2.8)$$

In the above description of the general sensor model, we have made the usual assumptions that there are no merged measurements or split measurements.

2.3 INFORMATION STRUCTURE

The information structure is the additional component which defines a distributed multitarget tracking and classification problem. Let N be the finite set of tracking agents (nodes). Let S_n be the set of sensors reporting to node n. We make the following assumptions on the S_n's:

a. $S = \bigcup_n S_n$,

b. $S_n \cap S_{n'} = \phi$ for $n \neq n'$. (2.9)

These assumptions state that the sensor sets for the various tracking nodes are mutually disjoint but collectively exhaust all the sensors.

The tracking nodes communicate to one another according to the communication schedules $\underline{C}$. $(t,n_1,n_2) \in \underline{C}$ means that node n_1 communicates to node n_2 at time t.

Let T be the time interval of interest and $(z,t,s) \overset{\Delta}{=} (z(k),k)$ be the data set from sensor s at t. Let $\underline{Z}$ be the set of all data sets and $\underline{K}$ be the set of all data set indices $(t, s) = k$. At any time t, the maximum information available to the entire system is given by the sets

$$Z_{|t} = \{(z,t',s) \in \underline{Z} | t' \leq t\} \qquad (2.10)$$

and

$$K_{|t} = \{(t',s) \in \underline{K} | t' \leq t\}. \qquad (2.11)$$

Because of communication constraints, the actual information available to a node s at time t is less. Consider the set of events when the information in the system changes (either through transmission or reception). The times when these events occur and the nodes (sensors or tracking nodes) which are affected are given below:

- sensor observation: $\underline{K}$
- reception of sensor data by a tracker:

  $\{(t,n) \varepsilon T \times N | (t,s) \varepsilon K, s \varepsilon S_n\}$
- transmission by a tracker:

  $\{(t,n) \varepsilon T \times N | (t,n,n') \varepsilon \underline{C}\}$
- reception by a tracker:

  $\{(t,n) \varepsilon T \times N | (t,n',n) \varepsilon \underline{C}\}$

To avoid unnecessary complexity, we assume (without loss of generality) that the four sets defined above are disjoint. Let $\underline{I}$ be the union of the four sets. A binary relation or a partial order $\prec$ can be defined on the set $\underline{I}$ as follows:

i. For $(n,t,t') \varepsilon N \times T \times T$, $(t,n) \varepsilon \underline{I}$, $(t',n) \varepsilon \underline{I}$ and $t < t'$ implies that

$$(t,n) \prec (t',n);$$

ii. $(t,s) \varepsilon \underline{K}$, $s \varepsilon S_n$ and $(t,n) \varepsilon \underline{I}$ implies that

$$(t,s) \prec (t,n);$$

iii. $(t,n,n') \varepsilon \underline{C}$ implies that

$$(t,n) \prec (t,n').$$

This binary relation or partial order on $\underline{I}$ thus satisfies all the constraints associated with perfect communication as defined by $\underline{C}$ as well as perfect memory at each processing node. $(\underline{I},\prec)$ characterizes the information flow in the system and is called the <u>information graph</u>. If all the sensor measurements (data sets) can be communicated perfectly through the communication network, a subset $Z(t,i)$ of $\underline{Z}$ (called the data base at (t,i)) for each node (t,i) in the graph $(\underline{I},\prec)$ can be defined by beginning with the minimal elements and following the rules shown below:

i. If (t,i) is a receiving node,

$$Z(t,i) = \cup\{Z(s,j) \mid (s,j) \rightarrow (t,i)\},$$

ii. If (t,i) is a transmitting node,

$$Z(t,i) = \begin{cases} Z(s,j) & \text{if } (s,j) \rightarrow (t,i) \\ \{(z(k),k)\} & \text{if } (t,i) = k \ \varepsilon \ \underline{K} \\ \phi & \text{otherwise.} \end{cases}$$

In the above $(s,j) \rightarrow (t,i)$ means that (s,j) is an immediate predecessor of (t,i) and $(z(k),k) \varepsilon \underline{Z}$ is the unique element whose second component is k.

With this construction of the data base, we see that $(t,i) \prec (s,j)$ if and only if $Z(t,i) \subseteq Z(s,j)$. Similarly, for each $(t,i) \varepsilon \underline{I}$, we can define the data index base $K(t,i)$, which correspond to the indices for the data sets in $Z(t,i)$.

Since there is a natural direction (along increasing time) in the graph, the arrowheads on the edges in a pictorial representation of the graph can be omitted. We would also omic those edges which are due to transitivity. From the graph, the flow of information in the system becomes very obvious. A node (t,i) is a parent of (s,j) if information flows from node i at time t to node j at time s.

Note that in the information graph, the receiving nodes correspond to the events when estimates have to be updated with the arrival of new information. For many applications, it is sufficient to use a reduced information graph, which is obtained by considering only these receiving nodes.

Thus the maximum information available to each information node (t,i) in the information graph is given by the data base $Z(t,i)$ or alternatively by the data index base $K(t,i)$. We are particularly interested in information nodes which correspond to the tracking nodes.

3. PROBLEM FORMULATION

Our objective is to consider a distributed version of multitarget tracking and classification problem as given in [5], [6]. In particular, we would like to evaluate the probability of each data association hypothesis using information communicated from other nodes. To this end, we shall first define the notions of tracks and hypotheses in this distributed framework.

3.1 TRACKS AND HYPOTHESES

We define the *measurement index set* $\underline{J}$ by

$$\underline{J} = \bigcup_{k \varepsilon \underline{K}} \{1,\ldots, N_m(k)\} \times \{k\}. \qquad (3.1)$$

An element (j,t,s) in $\underline{J}$ (called a *measurement index*) indicates that the j^{th} measurement in the data set generated by sensor s at time t. Any subset of $\underline{J}$ is called a *track* and any collection of nonempty tracks a *hypothesis*. A track is called possible if it contains at most one measurement index in each data set. A hypothesis is called possible if it contains only possible nonempty tracks and no two tracks in it intersect. Let $\mathfrak{H}$ and $\mathfrak{T}$ be the set of all possible hypotheses and tracks. When K is a subset of $\underline{K}$, define

$$J(K) = \{(j,k) \varepsilon \underline{J} | k \varepsilon K\}. \qquad (3.2)$$

Then J(K) is the measurement index set restricted by the data index base K. Similarly, if J is a subject of $\underline{J}$, define

$$\mathcal{T}(J) = \{ \tau \cap J \mid \tau \in \mathcal{T} \} \tag{3.3}$$

and

$$\mathcal{H}(J) = \{\{ \tau \cap J \mid \tau \in \lambda\} \setminus \{\phi\} \mid \lambda \in \mathcal{H}\}. \tag{3.4}$$

Thus $\mathcal{T}(J)$ is the set of all possible tracks defined on J and $\mathcal{H}(J)$ is the set of all possible hypothesis defined on J.

3.2 DISTRIBUTED FUSION PROBLEM

Consider the reduced information graph which is constructed from the information graph by picking out the tracking nodes. Each node in the graph is of the form (t,n) where t is a reception time (from a local sensor or other nodes) and n is a tracking node. Let J(t,n) = J(K(t,n)), i.e., the measurement index set available to agent n at time t. Assume that the information state for multitarget tracking and classification is given at each node, i.e., for each node (t,n), we have the following set of quantities:

$$\begin{gathered} \mathcal{H}(J(t,n)),\ (p(\lambda | Z(t,n)))_{\lambda \in \mathcal{H}(J(t,n))}, \\ \mathcal{T}(J(t,n)),\ (p(x(t) | Z(t,n), \tau))_{\tau \in \mathcal{T}(J(t,n))},\ \nu(K(t,n)) \end{gathered} \tag{3.5}$$

where $\nu(K)$ is the expected number of targets which are undetected in K and $p(x(t)|Z(t,n),\tau)$ is the probability distribution/density of $x(t)$ given the track τ and the data base $Z(t,n)$. To simplify the notation, we denote the above by the information node (t,n), i.e.,

$$\mathcal{H}(t,n),\ (p(\lambda|Z(t,n)))_{\lambda \in \mathcal{H}(t,n)},$$

$$\mathcal{T}(t,n),\ (p(x(t)|Z(t,n),\tau))_{\tau \in \mathcal{T}(t,n)},\ \nu(t,n) \qquad (3.6)$$

At each information node (t,n), the information state is to be updated. For a node corresponding to reception of sensor data at a tracking agent, the problem is straightforward and is the centralized multitarget tracking problem. For a node corresponding to reception of messages from other tracking agents, the problem is one of distributed fusion, i.e., to construct the information state using the information states from the predecessor nodes in the information graph. This problem can be interpreted as follows. Suppose the information states for multitarget tracking and classification (hypotheses, tracks, probabilities, etc.) are the messages communicated in the network. Each agent then tries to construct the results of the optimal tracker if the actual measurements were communicated through the network using only information states which can be communicated.

4. STATIONARY TARGETS WITH BROADCAST COMMUNICATION

In this section, we consider a special case to develop the basic results. These results would then serve as a basis for studying more complex situations. The target state is assumed to be stationary, i.e., $x_i(t) = x_i$ for all t. The communication is assumed to be the broadcast type, i.e.,

$$\underline{C} = \bigcup_{i=1}^{\infty} \bigcup_{n_1 \neq n_2} \{(t_i, n_1, n_2)\} \tag{4.1}$$

With this, the information graph and reduced information graph are given in Figure 1.

Consider two consecutive communication times t_b and t'_b.

Let

$$J \triangleq J(K_{|t_b}), \tag{4.2}$$

and

$$\overline{J} \triangleq J(K_{|t'_b}), \tag{4.3}$$

i.e., the cumulative measurement indices at t_b and t'_b respectively.

Also, define for each n in N,

$$J_n = \{(m,t,s) \varepsilon J | s \varepsilon S_n\} \cup \bar{J}. \tag{4.4}$$

Then J_n is the cumulative measurement index set available to each tracking agent n just before communication. K_n can be defined similarly. Let $K = K_{|t_b}$ and $\bar{K} = K_{|t'_b}$. Let Z_n, Z and $\bar{Z}$ be the cumulative measurements corresponding to J_n, J and $\bar{J}$ respectively.

We have to answer the following basic questions.

Q1. Given $\mathcal{T}(\bar{J})$, $\mathcal{H}(\bar{J})$, $(\mathcal{T}(J_n))_{n \in N}$, $(\mathcal{H}(J_n))_{n \in N}$, can we construct $\mathcal{T}(J)$ and $\mathcal{H}(J)$?

Q2. Given $(p(\lambda|\bar{Z}))_{\lambda \in \mathcal{H}(\bar{J})}$, $((P(\lambda|Z_n))_{\lambda \in \mathcal{H}(J_n)})_{n \in N}$, $(px(t_b)|\bar{Z},\tau)_{\tau \in \mathcal{T}(\bar{J})}$, $((p(x(t_b)|Z_n,\tau)_{\tau \in \mathcal{T}(J_n)})_{n \in N}$, $\nu(\bar{K})$ and $(\nu(K_n))_{n \in N}$, can we calculate $(p(\lambda|Z))_{\lambda \in \mathcal{H}(J)}$, $p(x(t_b)|Z,\tau)_{\tau \in \mathcal{T}(J)}$ and $\nu(K)$?

These questions will be considered separately.

4.1 HYPOTHESIS RECONSTRUCTION

We first address Question 1, which focusses on the construction of the global hypothesis from the local hypotheses.

Definition: Let $\tau \varepsilon \mathfrak{J}(J)$ be a track and $\bar{J} \subseteq J$, the restriction of τ onto $\bar{J}$ is defined as $\tau | \bar{J} = \tau \cap J$. $\bar{\tau} = \tau | \bar{J}$ is then a predecessor of τ and τ is a successor of $\bar{\tau}$. Similarly, when $\lambda \varepsilon \mathcal{H}(J)$ and $\bar{J} \subseteq J$, the predecessor $\bar{\lambda} = \lambda | \bar{J}$ is defined as

$$\lambda|J = \{\tau|J \ |\tau \in \lambda\}\setminus\{\phi\}. \qquad (4.5)$$

Then λ is the successor of $\bar{\lambda}$.

The following two lemmas are then obvious.

<u>Lemma 1</u>: Let J_1 and J_2 be two measurement index sets such that $J_1 \subseteq J_2$ and τ_1 be a track in a hypothesis $\lambda \in \mathcal{H}(J_1)$. Then for any successor λ_2 of λ_1, i.e., for any $\lambda_2 \in \mathcal{H}(J_2)$ such that $\lambda_2|J_1 = \lambda_1$, there exists a τ_2 in λ_2 such that $\tau_2|J_1 = \tau_1$. For given λ_2, such a track τ_2 is unique.

<u>Lemma 2</u>: Let J_1 and J_2 be two measurement index sets such that $J_1 \subseteq J_2$. Then for any $\lambda \in \mathcal{H}(J_2)$ we have

$$\text{Prob.}\{\lambda|J_1 \ |\lambda, J_2\} = 1 \qquad (4.6)$$

whenever $\text{Prob.}\{\lambda, J_2\} > 0$.

<u>Definition</u>: A hypothesis λ in $\mathcal{H}(J)$ is said to be <u>composed</u> of $(\lambda_n)_{n \in N} \in \prod_{n \in N} \mathcal{H}(J_n)$ if $\lambda|J_n = \lambda_n$ for all $n \in N$. The relationship is denoted by

$$\lambda \succ (\lambda_n)_{n \in N} \qquad (4.7)$$

An immediate property as a result of this definition is that, if λ is composed of $(\lambda_n)_{n \in N} \in \prod_{n \in N} \mathcal{H}(J_n)$, then all the λn's should share the same predecessor. Namely, we have

Lemma 3: If $\lambda \succ (\lambda_n)_{n \in N}$,

$$\lambda_n | \bar{J} = \lambda | \bar{J} \tag{4.8}$$

for all $n \in N$.

The proof is obvious from the fact that $\bar{J} \subseteq J_n$ for all $n \in N$.

A useful criterion for any $(\lambda, (\lambda n)_{n \in N})$ to be tested for composability is given by the following lemma.

Lemma 4: $\lambda \succ (\lambda_n)_{n \in N}$ if and only if, for any τ in λ, there exists a tuple

$$(\tau_n)_{n \in N} \in \prod_{n \in N} (\lambda_n \cup \{\phi\}) \tag{4.9}$$

such that $\tau = \bigcup_{n \in N} \tau_n$ and

$$\tau_n | \bar{J} = \tau | \bar{J}. \tag{4.10}$$

for all n in N.

Proof of Lemma 4: The "if" part is obvious. For "only if" part, suppose $\lambda \succ (\lambda_n)_{n \in N}$ and $\tau \varepsilon \lambda$. For each $n \in N$, let $\tau_n = \tau | J_n$. Then, since $\lambda_n = \lambda | J_n$, we have either $\tau_n \in \lambda_n$ or $\tau_n = \phi$. Since $J = \bigcup_{n \in N} J_n$, $\bigcup_{n \in N} \tau_n = \bigcup_{n \in N} \tau | J_n = \bigcup_{n \in N} \tau \cap J_n = \tau \cap J = \tau$. On the other hand,

$J \subseteq J_n$ implies that $\tau_n | \bar{J} = (\tau | J_n) | \bar{J} = \tau \cap J_n \cap \bar{J} = \tau \cap \bar{J} = \tau | \bar{J}$. Thus, $(\tau_n)_{n \in N}$ is an appropriate element which satisfies the lemma. ■

The following theorem forms the basis for hypothesis reconstruction.

<u>Theorem 1</u>: For any $\lambda \in \mathcal{H}(I)$ and any $(\lambda_n)_{n \in N} \in \prod_{n \in N} \mathcal{H}(J_n)$, $\lambda \succ (\lambda_n)_{n \in N}$ if and only if,

(1) for any τ in λ there exists a tuple $(\tau_n)_{n \in N} \in \prod_{n \in N} (\lambda_n \cup \{\phi\})$ such that

a. $\tau = \prod_{n \in N} \tau_n$ and

b. $\tau_n | \bar{J} = \tau | \bar{J}$ for all $n \in N$, and

(2) for any $n \in N$ and for any $\tau_n \in \lambda_n$ there exists a unique τ in λ such that $\tau | J_n = \tau_n$.

<u>Proof</u>:

The "only if" part is obvious from the definition of $\lambda \succ (\lambda_n)_{n \in N}$. The "if" part is as follows. Suppose (1) and (2) hold. (2) is equivalent to $\lambda_n \subseteq \lambda | J_n$ for all $n \in N$. On the other hand, (1) implies that $\lambda_n \supseteq \lambda | J_n$ as can be shown below. Let n be an arbitrary index in N and τ_n be an arbitrary track in $\lambda | J_n$. Then, by Lemma 1, there exists a unique extension τ in λ such that $\tau_n \subseteq \tau$ and $\tau | J_n = \tau_n$. For this τ, by (1) there exists a tuple $(\tilde{\tau}_{n'})_{n' \in N} \in \prod_{n \in N} (\lambda_{n'} \cup \{\phi\})$ such that

$\tau = \bigcup_{n' \in N} \tilde{\tau}_{n'}$ and $\tilde{\tau}_{n'}|\bar{J} = \tau|\bar{J}$ for each $n' \in N$. Then we have

$$\begin{aligned}
\tau|J_n &= \Big(\bigcup_{n' \in N} \tilde{\tau}_{n'}\Big)|J_n \\
&= \Big(\bigcup_{n' \in N} \tilde{\tau}_{n'}\Big) \cap J_n \\
&= \bigcup_{n' \in N} \tilde{\tau}_{n'} \cap J_n \\
&= \tilde{\tau}_n \cup \Big(\bigcup_{\substack{n' \in N \\ n' \neq n}} \tilde{\tau}_n \cap \bar{J}\Big) \\
&= \tilde{\tau}_n \cup (\tau|J) = \tilde{\tau}_n. \qquad (4.11)
\end{aligned}$$

Therefore, $\tau_n = \tau|J_n = \tilde{\tau}_n \in \lambda_n \cup \{\phi\}$. But, since $\tau_n \in \lambda_n$ (and hence $\tau_n \neq \phi$), $\tau_n \in \lambda_n$.

This theorem provides the following way of constructing $\mathcal{H}(J)$ and $\mathcal{T}(J)$.

1. For each $\bar{\lambda} \in \mathcal{H}(\bar{J})$, exhaust all the combinations of $(\lambda_n)_{n \in N} \in \prod_{n \in N} \mathcal{H}(J_n)$ such that $\lambda_n|\bar{J} = \bar{\lambda}$.

2. For each such $(\lambda_n)_{n \in N} \in \prod_{n \in N} \mathcal{H}(J_n)$,

   a. construct a unique extended track τ such that $\tau \cap \bar{J} = \tau$ by letting $\tau = \bigcup_{n \in N} \tau_n$ where τ_n is the unique extension in λ_n of

$\bar{\tau}$ for each $\bar{\tau} \in \bar{\lambda}$ and λ_{OLD} be the set of such tracks, and

b. exhaust all the combinations

$(\tau_n)_{n \in N} \in \prod_{n \in N} (\lambda_n \cup \{\phi\} \backslash \lambda_{OLD_n})$ (where $\lambda_{OLD_n} = \{\tau \cap J_n | \tau \in \lambda_{OLD}\}$) and construct new tracks $\tau = \bigcup_{n \in N} \tau_n$. This should be done in such a way that every τ_n in every λ_n is included in one of the tracks in the composite hypothesis. Let L_{NEW} be the set of all the hypotheses constructed in this way. Then, all the hypotheses λ in $\mathcal{H}(J)$ such that $\lambda \succ (\lambda_n)_{n \in N}$ is the set

$$\{\lambda_{OLD} \cup \lambda_{NEW} | \lambda_{NEW} \in L_{NEW}\}. \tag{4.12}$$

The construction of $\mathcal{J}(J)$ is obvious in the above description.

4.2 HYPOTHESIS EVALUATION

We now address Question 2, which focusses on the construction of the global probabilities of hypotheses and state distributions of tracks from the local values. We state the following lemma, derived in [5], [6], for the recursive evaluation of hypothesis.

<u>Lemma 5</u>: Consider an information node with cumulative measurements $\bar{Z}$ and cumulative measurement index set $\bar{J}$. Consider an immediate successor with cumulative measurements Z and cumulative measurement index set J. Let k = (t,s) be the most current data index. Then the recursive

evaluation of any λ in $\mathcal{H}(J)$ is given by

$$p(\lambda|Z) = C(Z)^{-1}\ p(\bar{\lambda}|\bar{Z})L_{FA}(k,\lambda) \prod_{\tau \in \lambda} L_k(Y(\tau,k),\tau) \tag{4.13}$$

where $Y(\tau,k)$ is the measurement for track τ in the current data set indexed by k, L_{FA} and $L_k(Y(\tau,k),\tau)$ are likelihood functions defined as follows:

<u>False Alarm Likelihood Function</u>

$$L_{FA}(k,\lambda) = n_{FA}(\lambda|k)!\ p_{N_{FA}}(n_{FA}(\lambda|k)) \prod_{j \in j_{FA}(\lambda|k)} p_{FA}(y_j|k) \tag{4.14}$$

where $n_{FA}(\lambda|k)$ is the number and $j_{FA}(\lambda|k)$ the set of false alarms in the current data set according to λ.

<u>Track-Measurement Likelihood Functions</u>

For previously detected targets which are detected again,

$$L_k(y,\tau) = \int p_m(y|x,k)p_D(x|k)p_{\bar{\tau}}(x)\mu(dx). \tag{4.15}$$

For previously detected targets which are missed,

$$L_k(y,\tau) = \int (1-p_D(x|k))p_{\bar{\tau}}(x)\mu(dx). \tag{4.16}$$

For newly detected targets

$$L_k(y,\tau) = \bar{\nu} \int p_m(y|x,k)p_D(x|k)q(x)\mu(dx). \tag{4.17}$$

In equations (4.15) and (4.17), y is the measurement associated with track τ, $p_{\bar{\tau}}(.) = p(x(t)|\bar{Z},\bar{\tau})$ is the state distribution/density for the track restricted by $\bar{Z}$ and $q(.) = p_{\phi} = p(x(t)|\bar{Z},\phi)$ is the density of undetected targets associated with Z.

A recursive application of Lemma 5 yields the following.

<u>Lemma 6</u>: Let Z be any data base, J the corresponding measurement index set and K be the data index base. Then for any $\bar{Z}$ such as $\bar{Z} \subseteq Z$ (with the corresponding $\bar{J}$ and $\bar{K}$),

(1) for each $\lambda \in \aleph(J)$,

$$p(\lambda|Z) = C(Z)^{-1} p(\bar{\lambda}|\bar{Z})\ \tilde{L}_{FA} \prod_{\tau\in\lambda} \tilde{L}_{\tau} \quad , \tag{4.18}$$

where C is a normalization constant,

$$\bar{\lambda} = \lambda|\bar{J}, \tag{4.19}$$

$$\tilde{L}_{FA} = \prod_{(t,s)\in K\setminus\bar{K}} L_{FA}(t,s,\lambda) \tag{4.20}$$

$$\tilde{L}_{\tau} = \begin{cases} \prod\limits_{(t,s)\in K\setminus\bar{K}} L_{(t,s)}(Y(\tau,t,s),\tau) & \text{if } \tau|J \neq \phi \\ \nu(\bar{K}) \prod\limits_{(t,s)\in K\setminus\bar{K}} L_{(t,s)}(Y(\tau,t,s),\tau) & \text{otherwise} \end{cases}$$

(4.21)

and

(2) for any $\tau \in \mathcal{H}(J) \cup \{\phi\}$,

$$p_{\tau} = \begin{cases} \prod_{(t,s)\in K\setminus\bar{K}} G_{\tau}(t,s)F(t,s)p_{\bar{\tau}} & \text{if } \bar{\tau} = \tau|\bar{J} \neq \phi \\ \prod_{(t,s)\in K\setminus\bar{K}} G_{\tau}(t,s)F(t,s)\bar{q} & \text{otherwise} \end{cases} \tag{4.22}$$

with $\bar{q} = p = \bar{p}(.|Z,\phi)$ and $\nu = \bar{\nu}\tilde{L}_{\phi}$. $G_{\tau}(t,s)$ and $F(t,s)$ are operators defined as

$$((G_{\tau}(t,s)(p))(x) = \frac{g_{\tau}(t,s)(x)p(x)}{\int g_{\tau}(t,s)(x)p(x)\mu(dx)}, \tag{4.23}$$

$$(F(t,s)(p))(x) = \int f_{\Delta t}(x|x')p(x')\mu(dx') \tag{4.24}$$

and

$$g_{\tau}(t,s)(x) = \begin{cases} p_m(y|x,t,s)p_D(x|t,s) & \text{for detected targets} \\ 1-p_D(x|t,s) & \text{for missed targets.} \end{cases} \tag{4.25}$$

Using Lemma 6, we can prove the following theorem.

<u>Theorem 2</u>: For stationary targets and broadcast communication, we have for every $\lambda \in \mathcal{H}(J)$,

$$p(\lambda|Z) = \tilde{C}^{-1}p(\bar{\lambda}|\bar{Z})^{-(\#N-1)}\Big(\prod_{n\in N} p(\lambda_n|Z_n)\Big)\prod_{\tau\in\lambda} \ell_{\tau} \tag{4.26}$$

where $\tilde{C}$ is a normalization constant, $\#N$ is the number of elements in N, and

$$\ell_\tau = \int \frac{\prod_{n \in N} \tilde{p}(x|Z_n,\tau)}{(\tilde{p}(x|\overline{Z},\tau))^{\#N-1}} \mu(dx) \tag{4.27}$$

The expected number of targets undetected up to K is:

$$\nu(K) = \int \frac{\prod_{n \in N} \tilde{p}(x|Z_n,\phi)}{(\tilde{p}(x|\overline{Z},\phi))^{\#N-1}} \mu(dx) \tag{4.28}$$

$\tilde{p}(x|Z_n,\tau)$ and $\tilde{p}(x|\overline{Z},\tau)$ are given by

$$\tilde{p}(x|Z_n,\tau) = \begin{cases} p(x|Z_n,\tau) & \text{if } \tau \neq \phi \\ \nu(K_n)p(x|Z_n,\phi) & \text{if } \tau = \phi \end{cases} \tag{4.29}$$

$$\tilde{p}(x|\overline{Z},\tau) = \begin{cases} p(x|\overline{Z},\tau) & \text{if } \tau \neq \phi \\ \nu(\overline{K})p(x|\overline{Z},\phi) & \text{if } \tau = \phi \end{cases} \tag{4.30}$$

$p(x|Z_n,\tau)$ and $p(x|\overline{Z},\tau)$ are the state distributions at the time of fusion conditioned by track τ, Z_n and $\overline{Z}$. Furthermore, the state distributions can be fused to obtain

$$p(x|Z,\tau) = \frac{\prod_{n \in N} p(x|Z_n,\tau)}{(p(x|\overline{Z},\tau))^{\#N-1}} \; C^{-1} \tag{4.31}$$

where C is a normalization constant.

<u>Proof</u> If we use the formula of Lemma 6 for λ, and $(\lambda_n)_{n \in N}$, Theorem 2 follows if the stationarity condition is assumed.

5. DYNAMIC TARGET MODELS WITH BROADCAST COMMUNICATION

Suppose the data index base after the current broadcast is K and the data index base after the last current broadcast is K.

Let

$$T_I = \{t \mid (t,s) \in K \backslash \bar{K}\} \tag{5.1}$$

and

$$x_I = (x_t)_{t \in T_I}. \tag{5.2}$$

Then the following theorem which can be proved readily holds for dynamic target models represented by a Markov process.

<u>Theorem 3</u>: For dynamic target models with broadcast communication, we have for every $\lambda \in \Lambda(J)$

$$p(\lambda|Z) = \tilde{C}^{-1} p(\bar{\lambda}|\bar{Z})^{-(\#N-1)} \Big(\prod_{n \in N} p(\lambda_n|Z_n) \Big) \prod_{\tau \in \lambda} \ell_\tau \tag{5.3}$$

where $\tilde{C}$ is a normalization constant and ℓ_π is the same as that in Theorem 2 with x replaced by x_I.

Note that this theorem states that the likelihood of track associations is now computed using the entire state trajectory over the interval defined by T_I instead of at just one time.

The following two corollaries are easy to show.

<u>Corollary 1</u>: Suppose T_I contains only one element, such as when broadcast communication is carried out at every time instant, then the evaluation formula of Theorem 3 holds with x_T reduced to $x(t)$ if the most recent communication time is t.

<u>Corollary 2</u>: Suppose $x_i(t)$ is a deterministic process, i.e., there exists a group of homeomorphic operators on X, $(\Phi_t)_{t \in (-\infty,\infty)}$ such that

$$f_{\Delta t}(x|x) = \delta(x - \Phi_{\Delta t}\ (\bar{x})) \qquad (5.4)$$

where $\delta(.)$ is the delta function on (X,μ). Then Theorem 3 holds with x_I replaced by $x_{\min(T_I)}$.

In the two special cases mentioned above, only the state distribution of the target at a single time is needed in evaluating the track association likelihoods. Otherwise, one would have to compute the distribution of the target state over an interval.

6. CONCLUSIONS

We have investigated the distributed multitarget tracking and classification problem. The approach is based on a Bayesian theory for centralized multitarget tracking and classification. Specific results are given for a case when the communication is of the broadcast type and algorithms for hypothesis formation and evaluation are presented for independent and identically distributed target models. The target dynamics can be both static or dynamic random processes.

REFERENCES

1. Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment," IEEE Trans. Automat. Contr., Vol. AC-23, pp. 618-626, Aug. 1978.

2. H.L. Wiener, W.W. Willman, I.R. Goodman and J.H. Kullback, "Naval Ocean-Surveillance Correlation Handbook, 1978," NRL Report 8340, Naval Research Lab., Washington, D.C., 1979.

3. I.R. Goodman, H.L. Wiener and W.W. Willman, "Naval Ocean Surveillance Correlation Handbook, 1979," NRL Report 8402, Naval Research Lab., Washington, D.C., 1980.

4. D.B. Reid, "An Algorithm for Tracking Multiple Targets," IEEE Trans. Automat. Contr., Vol AC-24, No. 6, pp. 843-854, Dec. 1979.

5. S. Mori, C.Y. Chong, E. Tse and R.P. Wishner, "Multitarget Multisensor Tracking Problems - Part 1: A General Solution and a Unified View on Bayesian Approaches," AI&DS Technical Memo, TM-1015-01, 1982.

6. S. Mori, C.Y. Chong, R.P. Wishner and E. Tse, "Multitarget Multisensor Tracking Problems: A General Bayesian Approach," Proc. 1983 American Control Conference, San Francisco, CA, 1983.

7. C.Y. Chong, S. Mori, E. Tse and R.P. Wishner, "Distributed Estimation in Distributed Sensor Networks," Proc. 1982 American Control Conference, Arlington, VA, 1982.

8. J.R. Delaney, R.T. Lacoss and P.E. Green, "Distributed Estimation in the MIT/LL DSN Test Bed," Proc. 1983 American Control Conference, San Francisco, CA, 1983.

9. R.R. Tenney and N.R. Sandell, Jr., "Detection with Distributed Sensors, "IEEE Trans. Aerospace and Electronic Systems, Vol. AES-17, No. 4, pp. 501-510, July 1981.

10. D. Teneketzis, "The Decentralized Quickest Detection Problem," Proc. 21st IEEE Conf. on Decision and Control, Fort Lauderdale, FL, 1982.

11. L.K. Ekchian and R.R. Tenney, "Detection Networks," Proc. 21st IEEE Conf. on Decision and Control, Ft. Lauderdale, FL, 1982.

12. E.C. Tacker and C.W. Sanders, "Decentralized Structures for State Estimation in Large Scale Systems," Large Scale Systems Theory and Applications, Vol. 1, No. 1, pp. 39-49, Feb. 1980.

13. J.L. Speyer, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem," IEEE Trans. Automat. Contr., Vol. AC-24. No. 2, pp. 266-269, April

1979.

14. C.Y. Chong, "Hierarchical Estimation," *Proc. 2nd MIT/ONR Workshop on Distributed Communication and Decision Problems*, July 1979.

15. A.S. Willsky, M. Bello, D.A. Castanon, B.C. Levy and G. Verghese, "Combining and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks," *IEEE Trans. Automat. Contr.*, Vol. AC-27, No. 4, pp. 799-813, Aug. 1982.

16. B.C. Levy, D.A. Castanon, G.C. Verghese and A.S. Willsky, "A Scattering Framework for Decentralized Estimation Problems," *Automatica*, Vol. 19, No. 4, pp. 373-384, July 1983.

17. D.A. Castanon and D. Teneketzis, "Nonlinear Data Fusion," *Proc. 5th MIT/ONR Workshop on C^3 Systems*, Monterey, CA, 1982.

18. R.L. Winkler, "Combining Probability Distributions from Dependent Information Sources," *Management Science*, Vol. 27, No. 4, pp. 479-488, April 1982.

19. V. Borkar and P. Varaiya, "Asymptotic Agreement in Distributed Estimation," *IEEE Trans. Automat. Contr.*, Vol. AC-27, No. 3, pp. 650-655, June 1982.

20. J.N. Tsitsiklis and M. Athans, "Convergence and Asymptotic Agreement in Distributed Decision Problems," *Proc. 21st IEEE Conf. on Decision and Control*, Ft. Lauderdale, FL, 1982.

21. C.Y. Chong, E. Tse, and S. Mori, "Distributed Estimation in Networks," *Proc. 1983 American Control Conference*, San Francisco, CA, 1983.

22. R.P. Hughes and R.R. Tenney, "A Multiobject Tracking Algorithm for a Distributed Sensor Network," *Proc. 3rd MIT/ONR Workshop on Distributed Information and Decision Systems*, Sept. 1980.

23. R.M. Tong, E. Tse and R.P. Wishner, "Distributed Hypothesis Formation in Sensor Fusion Systems," *Proc. 20th IEEE Conf. on Decision and Control*, San Diego, CA, Dec. 1981.

24. Y. Bar-Shalom, "On the Track-to-Track Correlation Problem," *IEEE Trans. Automat. Contr.*, Vol. AC-26, No. 2, pp. 571-572, April, 1981.

25. C.B. Chang and L.C. Youens, "Measurement Correlation for Multiple Sensor Tracking in a Dense Target Environment," *IEEE Trans. Automat. Contr.*, Vol. AC-27, No. 6, pp. 1250-1252, Dec. 1982.